Lappeenranta University of Technology
School of Engineering Science
Degree Program in Computer Science
Intelligent Computing Major

Master's Thesis

Lauri Rapo

GENERATING ROAD ORTHOIMAGERY USING A SMARTPHONE

Examiners: Prof., D.Sc. (Tech.) Lasse Lensu
M.Sc. (Tech.) Petri Hienonen

Supervisors: Prof., D.Sc. (Tech.) Lasse Lensu
M.Sc. (Tech.) Petri Hienonen

ABSTRACT

Lappeenranta University of Technology
School of Engineering Science
Degree Program in Computer Science
Intelligent Computing Major

Lauri Rapo

Generating road orthoimagery using a smartphone

Master's Thesis

2018

76 pages, 42 figures, 5 tables, 2 algorithms.

Examiners: Prof., D.Sc. (Tech.) Lasse Lensu
M.Sc. (Tech.) Petri Hienonen

Keywords: computer vision, road maintenance, orthophoto, inverse perspective mapping, structure from motion

Road surface maintenance is essential for transportation safety, but the task of manually gathering information on the condition of roads is time-consuming and laborious for maintenance personnel. This thesis proposes a system for generating road orthophotos from which the maintenance information can be extracted automatically. The only equipment required by the system consists of a smartphone and a vehicle. The approach was based on the inverse perspective mapping method, which required the automatic calibration of the camera. The automatic calibration was the main research problem of the thesis and was solved with the Structure from Motion technique. The performance of the camera calibration method is insufficient as estimates for the relative orientation of the camera—a critical part of the camera calibration—exceeded the acceptable error threshold 38% of the time on average. The camera calibration method requires further development to enable the orthophoto generation system to be put into use.

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
School of Engineering Science
Degree Program in Computer Science
Älykkään laskennan pääaine

Lauri Rapo

Tie-ortokuvien tuottaminen älypuhelimella

Diplomityö

2018

76 sivua, 42 kuvaa, 5 taulukkoa, 2 algoritmia.

Tarkastajat: Prof., TkT Lasse Lensu
Diplomi-insinööri Petri Hienonen

Hakusanat: konenäkö, tieverkon ylläpito, ortokuva, käänteinen perspektiivimuunnos, structure from motion

Teiden kunnon ylläpito on välttämätöntä liikenneturvallisuuden kannalta, mutta manuaalinen tiedonkeruu teiden kunnosta on aikaavievää ja työlästä. Tämä diplomityö esittelee järjestelmän, jolla voidaan luoda tie-ilmakuvia. Järjestelmän avulla kunnossapitotieto saadaan tuotettua automaattisesti näistä ilmakuvista. Ainoat järjestelmän vaatimat välineet ovat älypuhelin ja ajoneuvo. Järjestelmän toiminta perustui käänteiseen perspektiivimuunnokseen, joka edellytti kameran automaattista kalibrointia. Kameran automaattinen kalibrointi oli työn pääasiallinen tutkimusongelma, joka ratkaistiin Structure from Motion -tekniikalla. Kalibrointimenetelmän tarkkuus ei ole riittävä: Kameran suhteellisen asennon arviointi, joka on olennainen osa kalibrointia, ylitti hyväksytyn virheen rajan keskimäärin 38% ajasta. Kameran kalibrointimenetelmä vaatii jatkokehittämistä, jotta ilmakuvien tuotantojärjestelmä voitaisiin ottaa käyttöön.

PREFACE

I'd like to thank my supervisors, examiners, and Vionice Ltd. for making this work possible.

Lappeenranta, 10 May, 2018

Lauri Rapo

CONTENTS

1 INTRODUCTION **9**
1.1 Background . . . 9
1.2 Objectives and restrictions . . . 11
1.3 Structure of the thesis . . . 12

2 INVERSE PERSPECTIVE MAPPING **14**
2.1 Camera model . . . 14
2.2 Planar projective transform . . . 16
2.3 Virtual camera . . . 16

3 GENERATING ORTHOIMAGERY **20**
3.1 Mobile mapping systems . . . 20
3.2 Accelerometer calibration . . . 22
3.3 Vanishing point estimation . . . 23
3.4 Optical flow . . . 24
3.5 3D reconstruction . . . 26
3.5.1 Camera–road orientation . . . 29
3.5.2 Camera heading deviation . . . 31
3.5.3 Camera height . . . 31

4 3D RECONSTRUCTION PRE- AND POST-PROCESSING **32**
4.1 Sequence segmentation . . . 32
4.2 Frame masking . . . 32
4.3 Georegistration . . . 34
4.3.1 Affine transformation . . . 35
4.3.2 Gravity alignment of linear segments . . . 38
4.3.3 Altitude filtering . . . 39

5 SYSTEM IMPLEMENTATION **41**
5.1 Orthophoto composite generation . . . 41
5.2 Road surface features . . . 43
5.3 Camera positioning . . . 45
5.4 Image obstructions . . . 46
5.4.1 Static . . . 46
5.4.2 Dynamic . . . 48
5.5 System overview . . . 50

6 EXPERIMENTS AND RESULTS **51**
6.1 Approach and dataset 51
6.2 Parameters 54
6.3 Experiment results 54

7 DISCUSSION **61**
7.1 Intervariable correlations 61
7.2 System performance analysis 61
7.2.1 Error thresholds 62
7.2.2 Analysis of results 62
7.3 Future work 64

8 CONCLUSION **66**

REFERENCES **67**

APPENDICES
Appendix 1: Additional information on experiments
Appendix 2: Figures on orthophoto errors

LIST OF ABBREVIATIONS

BA	Bundle Adjustment.
CPU	Central Processing Unit.
FAST	Features from Accelerated Segment Test.
FRRN	Full-Resolution Residual Network.
GPS	Global Positioning System.
HSV	Hue Saturation Value.
IPM	Inverse Perspective Mapping.
LIDAR	Light Detection And Ranging.
LK	Lucas–Kanade.
LRIS	Laser Road Imaging System.
MMS	Mobile Mapping System.
OpenMVG	Open Multiple View Geometry.
RANSAC	RANdom SAmple Consensus.
SfM	Structure from Motion.
SIFT	Scale-Invariant Feature Transform.
SLAM	Simultaneous Localization And Mapping.
SVD	Singular Value Decomposition.
UTM	Universal Transverse Mercator.
VP	Vanishing Point.
WGS	World Geodetic System.

LIST OF SYMBOLS

β	Camera roll angle.
$\Delta\gamma$	Angle between the camera and vehicle heading about the y-axis.
γ	Camera heading angle in the world coordinate frame.
ω	Image scale ratio.
σ	Standard deviation.
θ	Angle between the camera and the road plane about the x-axis.
$\mathbf{c}$	Camera position vector.
d_p	The height of the physical camera from the road plane.
d_v	The height of the virtual camera from the road plane.
$\mathbf{E}$	Essential matrix.
$\mathbf{H}$	Homography matrix.
$\mathbf{H}_{\text{IPM}}$	Inverse perspective mapping matrix.
$\mathbf{K}$	Camera intrinsic parameters matrix.
$\mathbf{n}$	Normal vector.
$\mathbf{P}$	Camera projection matrix.
$\mathbf{R}$	Rotation matrix.
s	Transformation matrix uniform scale factor.
$\mathbf{S}_s$	Source point set.
$\mathbf{S}_t$	Target point set.
$\mathbf{T}$	Affine transformation matrix.
$\mathbf{t}$	Translation vector.

1 INTRODUCTION

This section introduces the background, motivation, objectives, and restrictions and summarizes the content of the rest of the thesis.

1.1 Background

Regular maintenance of road infrastructure is essential for ensuring transportation and social safety. Having knowledge of the current state of the road surface and its features is paramount, but the task of gathering information on roads manually on the field is time-consuming and laborious [1]. With orthoimagery of the road surface and computer vision, key surveillance and maintenance tasks, such as detecting worn out lane markings and sections of the asphalt which require repairs, could be automated. The automation of these tasks could ease the total workload of road maintenance personnel. Orthoimagery is an intuitive way of viewing road surface data, because the road and its features are inherently plane-like, as shown in Figure 1. The orthogonal viewpoint simplifies computer vision tasks in turn.

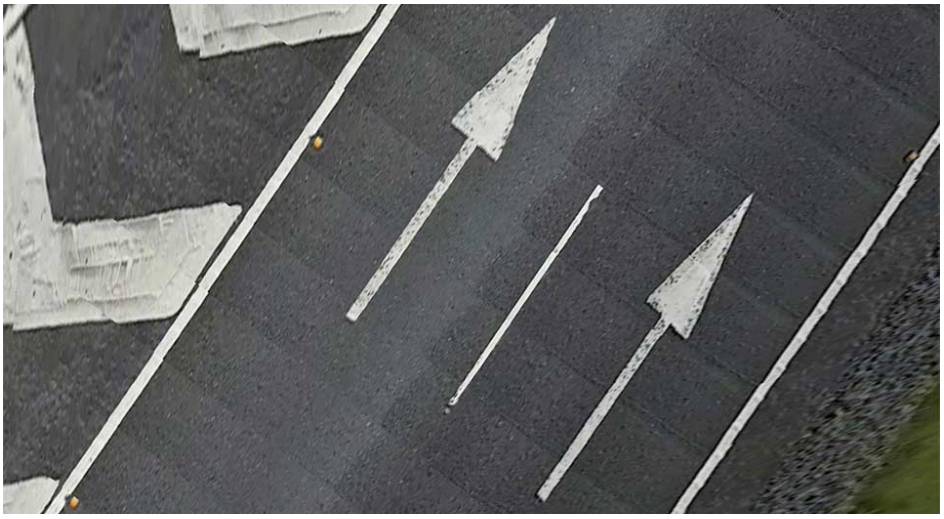


Figure 1. Road markings in an orthophoto composite.

Traditionally, orthoimagery has been collected using aircraft or satellites, but the approach has several drawbacks limiting its applicability for road inspections: Using aircraft or satellites for capturing orthoimagery can be expensive, which can set a limit on how often the information can be updated. This can lead to a situation where the information is outdated most of the time. The resolution of aerial or

satellite orthoimagery is often poor, which can make them useless in the context of road surface maintenance. An example of poor resolution can be seen in Figure 2a). Furthermore, when images are taken from a high altitude, objects above the road surface, such as trees, can obstruct the regions of interest. Figure 2b) presents an instance of this kind of obstruction.

a) **b)**

Figure 2. Google satellite imagery: a) A highway; b) Trees obstructing the view to the road.

Alternative methods for creating road orthophotos exists, however. There have been multiple instances where road orthophotos have been created by manipulating imagery captured using a front-facing vehicle-mounted camera employing the Inverse Perspective Mapping (IPM) method [2, 3, 4]. This approach does not suffer from the same shortcomings as traditional orthophotos: The resulting imagery can be of high resolution and updated regularly. Depending on the sensors used, the cost of creating road orthoimagery can be relatively low. The approach requires that the camera is calibrated appropriately for the task. Surveying vehicles with fixed pre-calibrated sensor rigs such as Mobile Mapping Systems (MMS) can be used, for example [4, 5].

This work was carried out in collaboration with Vionice Ltd., a Finnish technology company that specializes in utilizing computer vision for information production, asset management, and service solutions. The operating model of the company includes the goal of bringing computer vision to every vehicle. The company has opted to use smartphones for data collection and computer vision, making the usage of the aforementioned MMS's incompatible with the company's mode of operation.

MMS's can be costly and not available at all times, whereas it is far easier to install a smartphone to a vehicle's windshield. The IPM method is a prime candidate for orthophoto generation, but it should be implemented in a manner that is compatible with the company's mode of operation.

1.2 Objectives and restrictions

The primary objective of this work was to develop a robust system for the automatic generation of orthoimagery of the road surface. The system implementation includes an approach for the extraction of road features from the orthoimagery and their condition evaluation as a secondary objective. The system was developed following the objectives and restrictions brought upon by the operational model of Vionice Ltd., with the most distinguishable attributes being the ease of use and low associated costs. In practice, the solution needed to be implemented in a manner that only a smartphone (Android) and its sensors would be needed. These sensors include a monocular camera, a Global Positioning System (GPS) receiver, and a tri-axis accelerometer.

The generation of the orthoimagery would be carried out after the data has been collected with the smartphone and uploaded to a server, according to the company's practices. During the data collection, the smartphone would be installed to a vehicle's windshield and the camera would be facing approximately forward to the direction of movement. The data collection setup is presented in Figure 3.

Figure 3. The data collection setup: A smartphone installed to a car's windshield.

The chosen method for the generation of orthoimagery was the IPM transform, the use of which requires an appropriately calibrated camera. A simplified overview of the orthoimage generation system is presented in Figure 4. Since a smartphone can be installed in various ways to a vehicle, the camera would need to be automatically calibrated. The problem of the automatic calibration of the camera was consequently the main research objective and focus of this thesis. The chosen camera calibration method was experimented on, by assessing how accurately it estimates the variables used in the IPM transformation and orthoimage compositing.

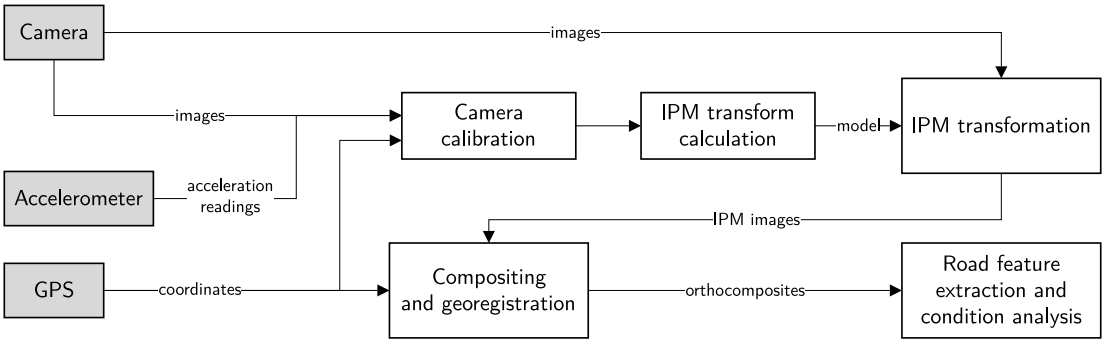


Figure 4. Simplified overview of the orthoimage generation system.

The following assumptions are applied to constrain the camera calibration problem:

- The camera is installed fixedly to the vehicle, i.e., the camera will not move relative to the vehicle during the data collection.
- The roll angle of the camera β (rotation about the axis of forward motion) is negligible, i.e., the camera installation is assumed level.
- The road surface can be interpreted as a plane in the proximity of the vehicle.
- A single IPM transform is enough to describe the projective relationship between the camera and the road surface for a video (no longer than 5 minutes).

1.3 Structure of the thesis

The rest of the thesis is structured as follows: In Section 2, the theory behind the IPM method is explored, which is the corner-stone for generating orthoimages from front-facing images. Section 3 focuses on the camera calibration for IPM and other methods for creating road orthoimagery. The section maps out various existing approaches and some novel ones, including the method that utilizes Structure from Motion (SfM), which was the chosen method for the definitive implementation of the IPM camera calibration.

In Section 4, factors that can improve SfM reconstruction results and the georegistration of the reconstruction are introduced. The information presented in the section can be valuable in other applications apart from the one of this thesis. Section 5 describes the implementation of the components that finalize the orthophoto generation system and views the system as a whole. The experiments and results are presented in Section 6. The experiment results are analyzed and possible future improvements are discussed in Section 7. The thesis is concluded in Section 8.

2 INVERSE PERSPECTIVE MAPPING

This section outlines the necessary theory behind the IPM transform, which is the basis for creating orthoimagery in the use case of this thesis. Section 2.1 introduces the used camera model, Section 2.2 sets out the theory on which the IPM is based on, and lastly Section 2.3 describes the formation of the IPM transform using a virtual camera.

2.1 Camera model

A camera is an enclosure with a lens, an opening, and a photosensitive surface [6]. Light focused by the lens enters the camera through the opening referred as the aperture and an inverted image is formed on the photosensitive surface. The functions of a camera can be approximated using the pinhole camera model [7], which is the model used throughout this work. The aperture is the size of a singular point in the pinhole camera model. A virtual image plane is generally positioned between the aperture and the camera's focal point so that a non-inverted image is formed. Lens distortions, which always occur with physical cameras to some degree, are not taken into account in the model since a lens is not considered to be a part of the model. Due to the aperture's infinitely small size and the lack of a lens, the camera model represents the ideal pinhole camera [6].

The camera model describes how 3D objects are projected on the 2D image plane, a process which is referred as perspective projection. The perspective projection is a linear mapping and can be expressed in matrix form. The projection is divided into two parameterizations, the intrinsic and extrinsic parameters. The intrinsic parameters model the camera's physical properties [8]. These properties include the focal length, pixel skewness, pixel aspect ratio, and the principal point of the image plane. The extrinsic parameters of the camera can be considered as the transformation that transforms the camera coordinate space into the world coordinate space [8]. In other words, the extrinsic parameters of the camera describe the position and orientation of the camera in the world coordinate space. This combination of position and orientation is often referred as a pose [9].

The used camera coordinate system is right-handed. The right, down, and forward axes are denoted by x, y, and z, respectively. The pinhole camera model is visualized in Figure 5.

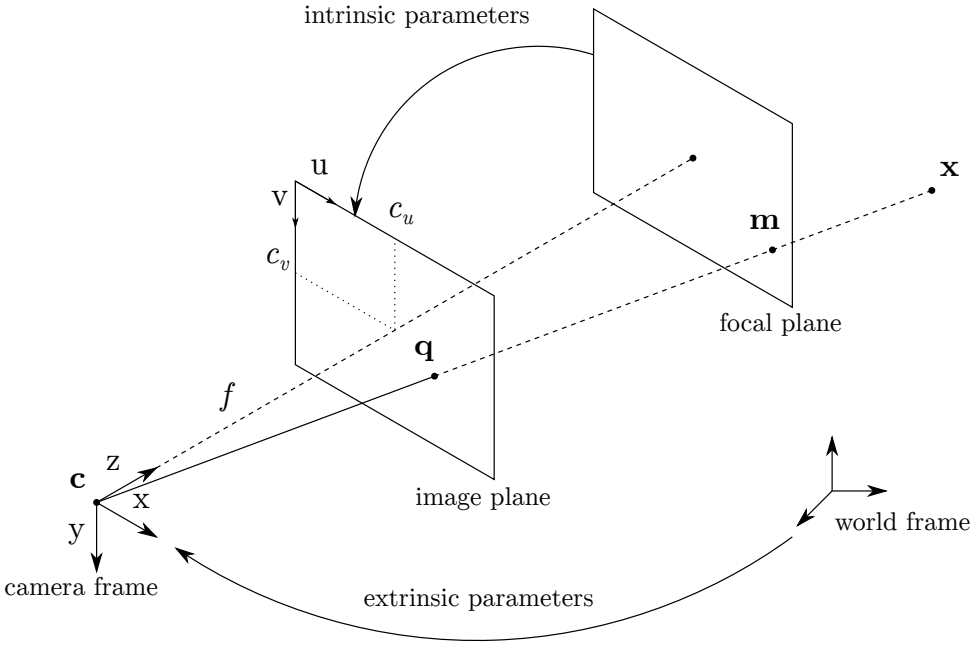


Figure 5. The pinhole camera model.

If we combine the intrinsic and extrinsic camera parameters, the camera projection matrix $\mathbf{P}_{3\times4}$ can be formulated:

$$\mathbf{P}_{3\times4} = \mathbf{K}\left[\mathbf{R} \;\middle|\; \mathbf{t}\right] = \underbrace{\begin{bmatrix} f & 0 & c_u & 0 \\ 0 & f & c_v & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} & \mathbf{R}_{3\times3} & & \mathbf{t}_{3\times1} \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

In Equation 1, the 3×4 matrix $\mathbf{K}$ is called the intrinsic matrix, which contains the intrinsic camera parameters. The latter is called the extrinsic matrix, containing the extrinsic parameters. In the intrinsic matrix, f denotes the camera focal length. Scalars c_u and c_v denote the horizontal and vertical components of the principal point of the image plane. This intrinsic matrix is simplified with the assumptions that the image pixels are of square aspect ratio and the pixels are not skewed. The intrinsic parameters of the camera $\mathbf{K}$ are assumed known, as the necessary information is retrieved from the smartphone's metadata. The 4×4 extrinsic matrix consists of the camera rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$. The translation vector does not represent the location of the camera, but the position of the origin of the world coordinate system, expressed using the camera coordinate system. The translation vector can be calculated with $\mathbf{t} = -\mathbf{R}^\top\mathbf{c}$, where $\mathbf{c}$ is the camera position in the world coordinate system.

2.2 Planar projective transform

The frames recorded by the monocular camera of the smartphone are 2D projections of the physical 3D scene. From a single frame, it is impossible to recover the original scene due to the loss of one dimension. However, if the road surface is viewed as a plane, in a sense it is possible to recover this plane and project it in any way we wish. This process is known as IPM [3, 10, 11], which is in itself an application of the planar projective transform. The planar projective transform is defined as a linear transformation by a non-singular 3×3 matrix on homogeneous vectors [12]:

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{2}$$

which can also be expressed more compactly as $\mathbf{x}' = \mathbf{H}\mathbf{x}$, where $\mathbf{H}$ is referred as a homography matrix. If we denote $\mathbf{x}$ as a point on the road plane and $\mathbf{x}'$ as the projection of the point on the image plane, the points are related by the homography matrix $\mathbf{H}$, which has 8 degrees of freedom. Using the homography matrix $\mathbf{H}$, a planar object can be transformed as if it was viewed from another perspective, without losing any information in the process.

2.3 Virtual camera

If an orthographic projection of the road surface is required, a homography can be calculated for the purpose. Let us assume that the homography we seek represents how the road plane is projected to a virtual camera which is facing directly downwards. Lines which are parallel in reality appear to converge to the Vanishing Points (VP) of the scene. After the virtual camera homography is applied to a camera frame, these lines become visibly parallel.

Let us denote a set of points on the road plane as $\mathbf{Q}_r$ in homogeneous coordinates. The projections of these points on the image plane of the physical camera are $\mathbf{Q}_p$ and on the virtual camera $\mathbf{Q}_v$ through homographies $\mathbf{H}_p$ and $\mathbf{H}_v$, respectively. The projective relationships can be expressed as follows:

$$\mathbf{Q}_p = \mathbf{H}_p \mathbf{Q}_r \tag{3}$$
$$\mathbf{Q}_v = \mathbf{H}_v \mathbf{Q}_r. \tag{4}$$

By employing Equations 3 and 4, we can express the points projected to the virtual camera using the points projected on the physical camera and the two homographies:

$$\mathbf{Q}_v = \mathbf{H}_{\text{IPM}} \mathbf{Q}_p, \text{ where } \mathbf{H}_{\text{IPM}} = \mathbf{H}_v \mathbf{H}_p^{-1}. \tag{5}$$

This leaves the task of finding homography $\mathbf{H}_{\text{IPM}}$ which maps the points on the physical camera directly to the virtual camera, which can be referred as the IPM. The IPM transform can be defined using the variables listed in Table 1 with the following formula [2, 12]:

$$\mathbf{H}_{\text{IPM}} = \mathbf{K} \cdot \left(\mathbf{R} + \frac{\mathbf{t}^\top \mathbf{n}}{d_p} \right) \cdot \mathbf{K}^{-1}. \tag{6}$$

Table 1. Variables used in the virtual camera IPM transformation.

Variable	Symbol	Dim.
Intrinsic parameter matrix of the camera	$\mathbf{K}$	$\mathbb{R}^{3\times3}$
Relative rotation between the physical and virtual camera	$\mathbf{R}$	$\mathbb{R}^{3\times3}$
Relative translation between the physical and virtual camera	$\mathbf{t}$	$\mathbb{R}^{3\times1}$
Normal vector of the road plane	$\mathbf{n}$	$\mathbb{R}^{3\times1}$
Camera height from the road plane	d_p	$\mathbb{R}^{1\times1}$

The virtual camera scenario is presented in Figure 6, where $\mathbf{c}_p$ and $\mathbf{c}_v$ present the locations of the physical and virtual camera, respectively. The road plane is denoted by P and its normal vector by $\mathbf{n}$. The heights of the cameras from P are denoted by d_p and d_v using the same convention as previously. The extrinsic parameters of the virtual camera are defined by the translation vector $\mathbf{t}$ and rotation matrix $\mathbf{R}$. If we denote t_z and t_y as the forward and upward translations, respectively, then $\mathbf{t} = \begin{bmatrix} 0 & t_z & t_y \end{bmatrix}^\top$.

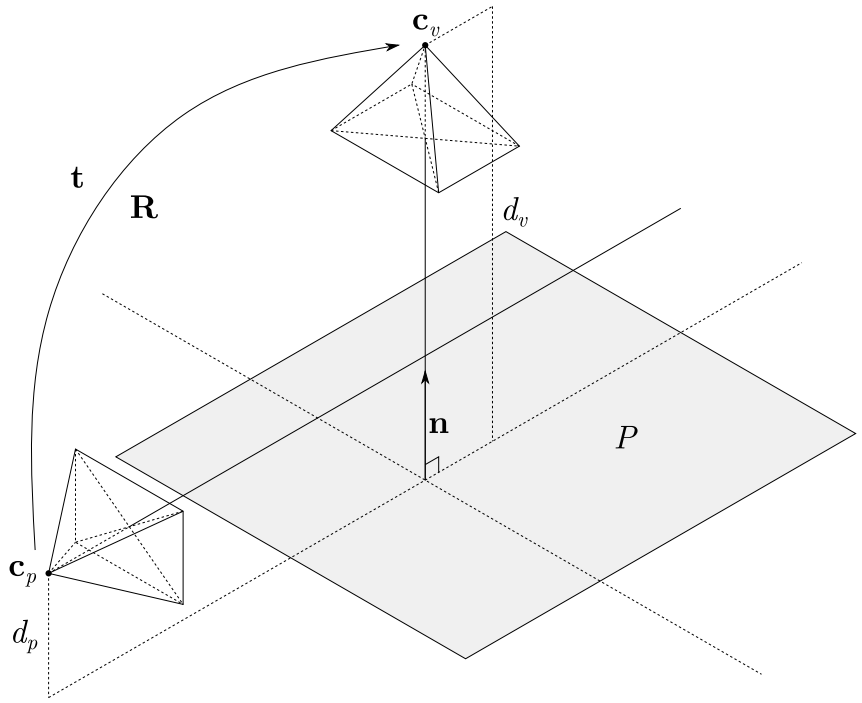


Figure 6. Visualization of the physical and virtual camera.

The relative pose of the virtual camera can be regarded as static. Since the intrinsic parameters of the camera are known, the variables which remain unknown in Equation 6 are the plane normal vector $\mathbf{n}$, the camera height d_p, and the rotation matrix $\mathbf{R}$. These unknown parameters are the starting point of the camera calibration problem. The search for the normal vector $\mathbf{n}$ is essentially the search for the orientation difference between the physical camera and the road surface, which will be also referred as the camera–road orientation from now on.

Including the assumption that the camera roll angle β is negligible, the estimation of the normal vector $\mathbf{n}$ can be simplified to finding the angle difference between the camera and road surface along the x-axis, denoted by θ. When angle θ is known, the normal vector can be calculated with $\mathbf{n} = \begin{bmatrix} 0 & \cos(\theta) & \sin(\theta) \end{bmatrix}$. Angle θ is visualized in Figure 7a) along with d_p. The rotation matrix $\mathbf{R}$ can be defined to apply a 90 degree rotation about the x-axis, but the rotation about the y-axis has to be considered as well: The heading of the physical camera and the recording vehicle are often not equal, which needs to be taken into account, to ensure accurate IPM image compositing. The angle difference between the camera and the vehicle about the y-axis, denoted by $\Delta\gamma$, should be included in $\mathbf{R}$. From now on, $\Delta\gamma$ will be also referred as the camera heading deviation. The formation of the angle $\Delta\gamma$ is visualized in Figure 7b). Examples of the IPM transform being used on images are shown in Figure 8.

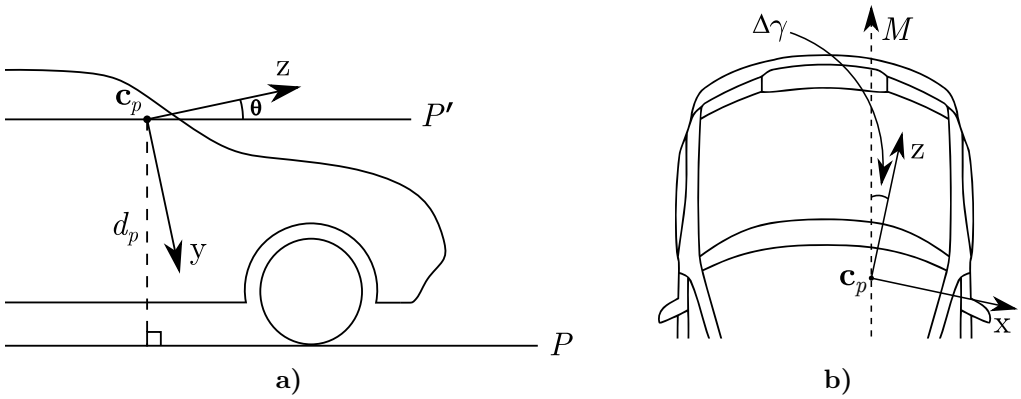


Figure 7. System component variables visualized: a) Side view of the setup. The physical camera's position is denoted by $\mathbf{c}_p$ and the road plane by P. P' is P translated to the level of $\mathbf{c}_p$ so that the formation of angle θ becomes clear. b) Top view of the setup. The angle $\Delta\gamma$ is formed by the z-axis of the camera and the movement direction of the vehicle, which is denoted by M.

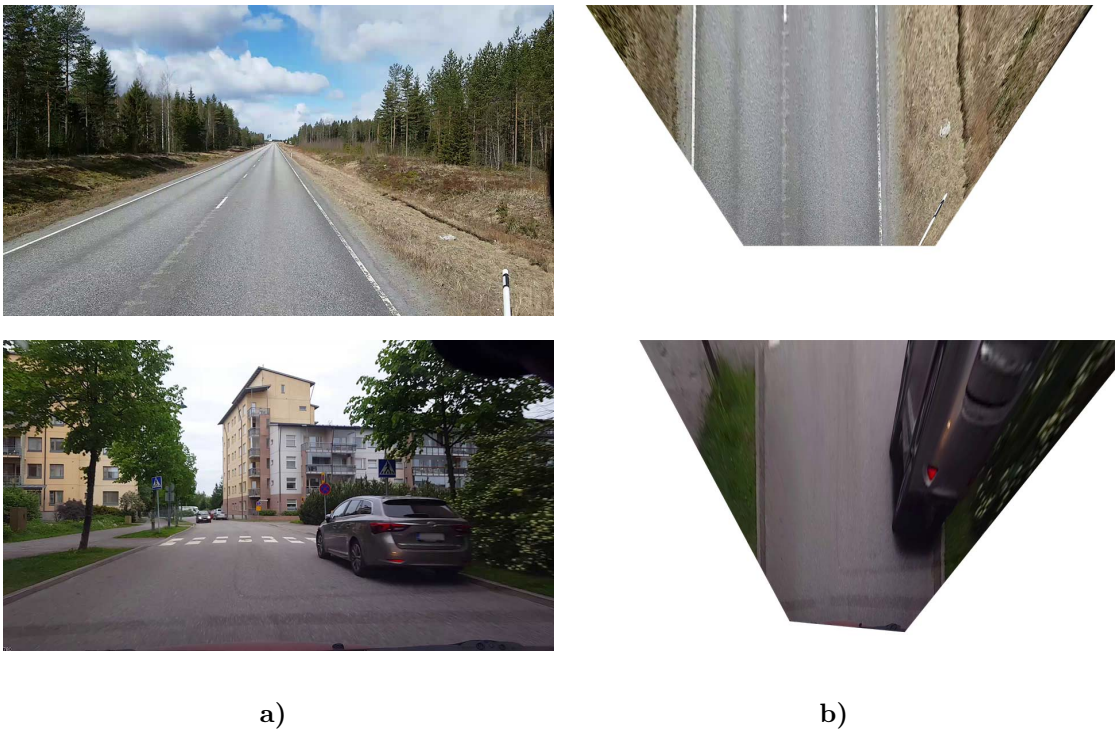


Figure 8. IPM in operation: a) Camera frames; b) Corresponding inverse-perspective frames. Notice how the car (a non-planar object) appears in the second IPM image.

3 GENERATING ORTHOIMAGERY

This section outlines methods for generating road orthoimagery and obtaining the necessary information needed by the IPM transform. These methods can be divided into two categories: Methods requiring user interaction to some extent, such as the manual calibration of the camera (manual), and methods requiring little or no user involvement (automatic). The manual methods are covered in Sections 3.1 and 3.2 and the automatic methods in the subsequent sections.

The manual methods include some approaches that are incompatible with the thesis' use case due to the required resources or the mode of operation. These methods are presented because they can be useful for parties in different circumstances. The automatic methods focus solely on the problem of camera calibration for the IPM transform and were selected on the basis of their compatibility with the resources available in the thesis' use case.

The methods outlined in Sections 3.2, 3.4, and 3.5 are methods devised by the author and the rest originate from related literature unless stated otherwise. It should be noted that from all of the automatic camera calibration methods, the method outlined in Section 3.5 was the only method capable of estimating the camera height from the road surface automatically. The method was chosen as the definitive solution for the IPM camera calibration problem.

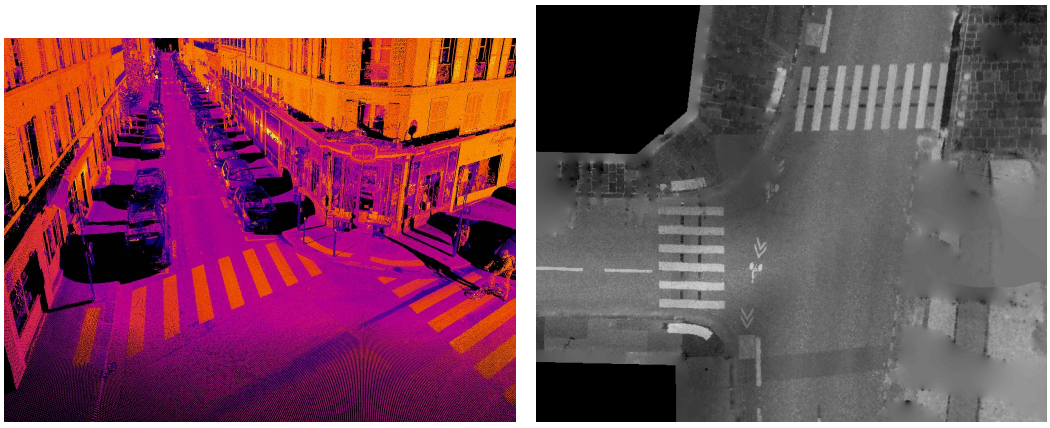
3.1 Mobile mapping systems

One of the more robust ways of generating orthoimagery of the road is the usage of MMS's, due to accurately calibrated sensor rigs. The rigs usually consist of sensors such as cameras, accelerometers, gyroscopes, gravimeters, Light Detection And Ranging (LIDAR) scanners, and GPS receivers. The larger the variety and the accuracy of the sensors, the potentially more accurate the image generation results.

In the research by Yagishita and Chikatsu [13], an approach is detailed where orthophotos are generated using a camera and a laser scanner. The sensor rig is pre-calibrated and presumably the system uses a single IPM transform for warping the imagery. The novelty factor in the study comes from the manner the laser scanner is used: Shadows in the resulting orthophotos can reduce visibility and overall image information content. However, shadows have little effect on laser intensity

readings from the scanner, allowing for the removal of shadows from the images through brightness correction.

Laser scanners have been used for orthophoto generation with a different approach as well, as has been presented in the work by Vallet and Papelard [14]. In their approach, the LIDAR point cloud is used directly to generate orthoimagery of the road surface. In addition, a digital terrain model of the road is created, while points that are not part of the ground are filtered out. Due to the gaps of varying size in the point cloud, the authors used Poisson interpolation to obtain a continuous raster image from the original reflectance values. The produced orthoimages are in grayscale since laser scanners typically can only measure reflection intensity. Sensor fusion could potentially resolve this shortcoming by registering the point cloud with camera images [15, 16, 17]. A laser-scanned point cloud and a generated orthophoto are shown in Figure 9.



a) **b)**

Figure 9. Images from the research by Vallet and Papelard [14]: a) Laser-scanned point cloud, with the color presenting reflectance; b) Poisson-interpolated grayscale orthophoto.

Oliveira and Correia have presented a framework for the automatic detection and classification of cracks in the road surface [18]. The road cracks are detected from orthophotos generated using the Laser Road Imaging System (LRIS) which consists of two high-resolution line scan cameras and two high-power lasers. The cameras and the lasers are located in the back of the surveying vehicle and point down towards the road surface. Due to the consistent illumination provided by the lasers, the system is unaffected by variations in the outside lighting conditions and shadows.

Meguro et al. have proposed a method for generating orthoimages of the road by using a single-frequency GPS, speed and yaw-gyroscope sensors, accompanied by

a camera [4]. The orthoimage generation is based on the IPM method. In their approach, the relative position of the camera is estimated beforehand, and a single fixed homography is being used during the recording. A similar execution can be seen in the work by Yang et al., where the used sensors include two cameras, a GPS receiver, and an inertial measurement unit [5]. The camera calibration is carried out after the installation of the cameras, which should not move relative to the vehicle during the data collection.

3.2 Accelerometer calibration

A semi-automatic method for calibrating the camera for the IPM transform was devised by the author during the development of the orthophoto generation system. The strength of the approach lies in the fact that it requires no technical skills and can be carried out by anyone, given that an adequate software implementation with proper instructions exists. The method is based on making use of the tri-axis accelerometer of the smartphone. It is required that the camera does not move relative to the vehicle during recording, as the method provides a one-off calibration.

The calibration process begins with the user laying the smartphone flat on the ground at the four corner points around the used vehicle. At each corner, a reading is saved from the device's accelerometer. When the phone is at rest on the ground, the accelerometer provides a good estimate of the device's gravity vector. The gravity vector can be interpreted as the orientation of the device. From these four measurements, we can calculate the mean orientation vector, which approximates the normal vector of the road surface around the vehicle.

Finally, the phone is installed to the vehicle and a final accelerometer reading is recorded. When the last accelerometer measurement is done, the driver should sit in the driver's seat, since the orientation of the vehicle can change depending on the weight distribution. When we calculate the orientation difference between the road plane normal vector and the gravity vector of the phone's final position, we acquire an estimate for the relative orientation between the camera and the road plane. In the context of a fleet of recorders, the method seemed a bit too cumbersome to be practical, which caused the technique to be discarded during the ideation phase.

3.3 Vanishing point estimation

The VP of the scene can be used to estimate the relative orientation of the camera given the assumption that the field of view is known. An image can have up to three VP's, the dominant one being closest to the image center. By determining the location of the dominant VP, the relative camera rotations about the x- and y-axes can be determined, which then could be used as estimates for θ and $\Delta\gamma$, respectively. If required, the roll angle β would need to be determined by other means, as its estimation is not possible using a single VP.

In the work by Kheyrollahi and Breckon [19], a method for estimating the location of the dominant VP is presented and is then used to determine the orientation of the camera for the IPM transform as a one-time calibration. On top of the orthoimage generation, the authors also present a solution for the automatic detection and classification of road markings. The VP detection algorithm is based on line intersections, where the lines are obtained by the Hough transform [20]. First, Canny edge detection [21] is applied to the images. Then the images are preprocessed by a temporal filter, in order to reduce errors caused by textures and other obstructions. After the filtering, line-like image features which seem to intersect at the VP are more prominent, as can be seen in Figure 10.

a) **b)**

Figure 10. Effects of temporal filtering [19]: a) Output of Canny edge detection for a single frame; b) Result of temporal filtering on a Canny edge image sequence.

From the temporally filtered image, the lines can be detected using clustering in the Hough space. The intersection points are then calculated for all possible combinations of lines. The resulting points are then clustered with k-nearest neighbor clustering, with $k = 3$ as no more than three VP's can be present in a single image. The point clusters are given a score as follows:

$$\Psi(U) = \sum_{i=1}^{n}(|x_c - x_i| + |y_c - y_i|). \quad (7)$$

The score for cluster U is the sum of the Manhattan distances between the VP of the previous frame (x_c, y_c) for all the intersection points in U. The centroid of the lowest scoring cluster is selected as the VP of the frame, after it has been averaged using the VP of the preceding frame. The authors estimate that the VP estimation converges after about 100 frames.

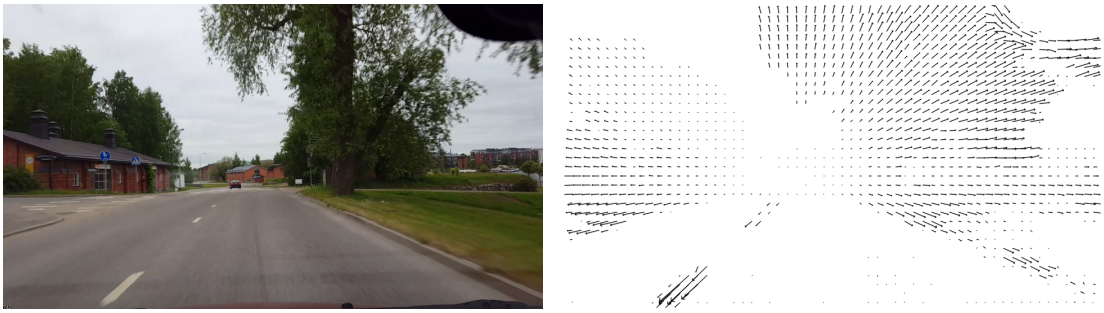
Nieto et al. have also proposed a method for the camera calibration for the IPM transform using VP estimation [22]. In their approach, the VP estimation is also based on lines obtained with the Hough transform. However, these lines are based on temporally filtered road lane markings, which are obtained through histogram-based segmentation of the original images. The intersection point of the lines is solved by an overdetermined system of equations using Singular Value Decomposition (SVD). The obtained vanishing points are then stabilized using a low-pass filter, reducing the errors caused by possible outlier VP's.

The presented VP detection methods include some factors which can cause problems: Canny edge detection requires two threshold parameters, and may produce poor results if the parameters are chosen poorly. Parameter configurations which work well on a specific video may not function at all for another video if the lighting or other conditions are different enough. Multiple methods exist for finding suitable parameters for Canny edge detection automatically [23, 24, 25], but there is no guarantee that they will work in all conditions. The detection of Hough lines suffers from the same problems, as it requires some parameters to function. The method proposed by Nieto et al. will not function in a gravel road scene, where there is no lane markings present since the method is based on them. An assumption, that lane markings would be always present, cannot be made in the use case of this thesis.

3.4 Optical flow

Optical flow is a 2D vector field, which is used to estimate the apparent motion in a pair of images by directly using the change in pixel intensities [26]. In its basic form, the problem of determining the flow is an underconstrained problem. The Lucas–Kanade (LK) optical flow algorithm is based on the assumption that the flow

is constant in a small neighborhood of pixels. These pixel neighborhoods contain enough information to make the problem not underconstrained and the flow can then be determined [26]. The LK optical flow can be used for the tracking a sparse feature set, for example, corner points. A dense variant of optical flow has been proposed by Farnebäck, where the flow is calculated for all pixels in the images [27]. An example of the dense optical flow is presented in Figure 11.



a) b)

Figure 11. Dense optical flow: a) Frame from an image sequence; b) Optical flow vector field.

The optical flow vector field can be used to estimate the camera orientation relative to the road surface in at least two ways. The first approach is based on the estimation of the main VP in an image sequence using the vector field. When calculating the dense optical flow between a pair of images, the result can contain errors caused by large pixel displacements, lack of texture, or erratic camera vibrations. When more image pairs are used, a temporal filter can be applied to the vector field, which reduces the effect of the errors present in individual image pair vector fields.

After a sufficient amount of frames, the filtered vector field represents a good estimate for the average scene motion. From the filtered vector field, the position of the main VP can be estimated by interpreting the field vectors as lines. In a successful case, the lines seem to intersect at a common point, which can be interpreted as an estimate for the VP. Using methods presented in Section 3.3, the position of the point can be calculated and then used to determine the relative orientation of the camera.

The second approach is based on the LK method and Nistér's five-point algorithm [28]. For a pair of frames, a set of corner points is calculated for the first frame using Features from Accelerated Segment Test (FAST) [29], for example. The positions of the corner points are tracked to the second frame using LK optical

flow. The matching points are then used to compute the essential matrix **E** [12] between the two views with the five-point algorithm. From the essential matrix **E**, the extrinsic matrix, i.e., the pose of the second camera, can be extracted using SVD [12].

Using the rotation matrix **R** and camera position vector **c** obtained from the extrinsic matrix, the difference between the orientation and actual movement direction of the camera can be estimated. Using this information, the orientation of the camera relative to the road can be approximated, if we assume that the relative camera movement matches the movement of the vehicle since the camera is fixedly installed. As in the previous method, a temporal filter can be applied to the retrieved camera orientations to obtain a more robust result. A more detailed description of this method can be found in the author's previous work [30].

3.5 3D reconstruction

3D reconstruction is an application of photogrammetry, a science in which geometric and semantic information is extracted from images [31]. One approach for 3D reconstruction is the SfM technique [32]. SfM computes a camera pose for each of the input images and a point cloud which represents the global structure of the scene. The relative camera poses are calculated by matching image features between input images and the scene structure is calculated by triangulating the 3D positions of the image features. A high-level visualization of the SfM technique is presented in Figure 12. Another possible approach for 3D reconstruction is the Simultaneous Localization And Mapping (SLAM) method, in which the camera pose and the scene structure are estimated online [33, 34] whereas in SfM this is done offline. SLAM algorithms have been developed real-time applications in mind, meaning that the focus is primarily on performance.

Having all of the scene's images as a starting point has its inherent advantages and processing time was not an issue, as real-time performance was not required. These factors steered the development towards SfM utilization. SfM is used extensively in other applications of Vionice, making the use of it in the orthophoto generation even more suitable: Reconstruction results which have been calculated previously for different purposes can be reused in the orthophoto generation. On top of that, by using SfM we can retrieve all the information required by the IPM method (camera height, relative orientation, and absolute heading of the camera) meaning that

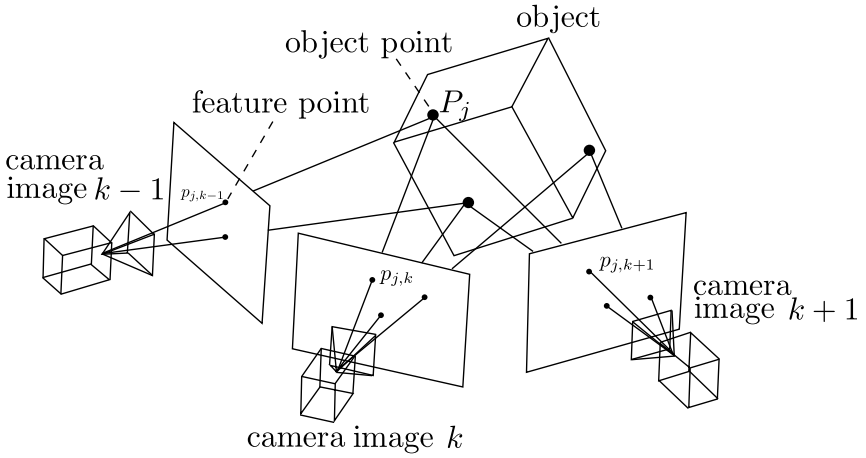


Figure 12. Structure from Motion [35].

additional calibration techniques would not be needed. The specific implementation of SfM being used was the one found in the Open Multiple View Geometry (OpenMVG) software library [36].

OpenMVG provides two methods for solving the camera positions and orientations: global and incremental. The global method [37] attempts to solve all the camera poses in the scene simultaneously while the incremental one [35] starts to build the reconstruction from an initial pair of frames, adding more frames one by one. The incremental method can automatically find candidates for the initial pair of frames [35], but this entails some inherent problems: The quality of the reconstruction can be highly dependent on which image pair is selected for the initialization of the reconstruction and on the order in which the rest of the images are added [38]. The incremental method can be slow and subjected to drift [39], i.e., errors in the pose and structure estimation accumulate over time while new frames are added. By contrast in the global method, the errors are distributed evenly across the reconstruction [37]. The global pipeline was the method of choice for the implementation of the solution due to the presented disadvantages regarding the incremental method.

Algorithm 1 presents the global SfM algorithm [37] at a high level starting from the feature extraction phase. The first step in the SfM algorithm is the extraction of feature points from the input images. Usually, these features consist of corner points, where the image gradient intensity is high and trackable. Then, the features extracted from the images are matched with features from other images using information in the feature descriptors in order to find correspondences. The decision which images will be matched to which can be based on image proximity or succession, for example. It is also possible to match all images with each other, but this

can be computationally expensive. After the matches have been computed, they can be filtered to ensure they match a certain mathematical model, such as the essential matrix.

Algorithm 1: Sketch of the global SfM algorithm

input : Image sequence
Camera intrinsics
output: 3D point cloud
Camera poses
Extract features from each image
Match features between images
Filter out false matches
Compute the relative pairwise camera rotations
Compute the global camera rotation
Compute the relative camera translations
Compute the global camera translation
Compute the global structure by triangulation
Refine the structure, camera poses, and camera parameters with Bundle Adjustment (BA)

At this point, the estimation of the camera poses and the global structure begins. The relative pairwise camera rotations are estimated first, followed by the global rotation estimation. Next, the relative translations are estimated followed by the global translation estimation. After the camera poses are determined, the structure of the scene can be reconstructed using triangulation of the feature points. The final optional, but a recommended step, is BA, where the scene structure, camera poses, and camera parameters are optimized to reduce errors through a non-linear least-squares algorithm. The Levenberg–Marquardt algorithm [40] has become a popular choice in this application [41].

Once the reconstruction of a scene is complete, a point cloud of the environment and the camera poses for each used image frame are retrieved. An example of a point cloud created using OpenMVG is presented in Figure 13. Initially, the coordinate system of the reconstruction has no connection to the real world, as only the relative scale of objects is attained. Through the process of georegistration, the detailed description of which can be found in Section 4.3, the point cloud and the camera poses can be tied to the real world. After the reconstruction has been georegistered, we can measure absolute distances in the coordinate space, for example.

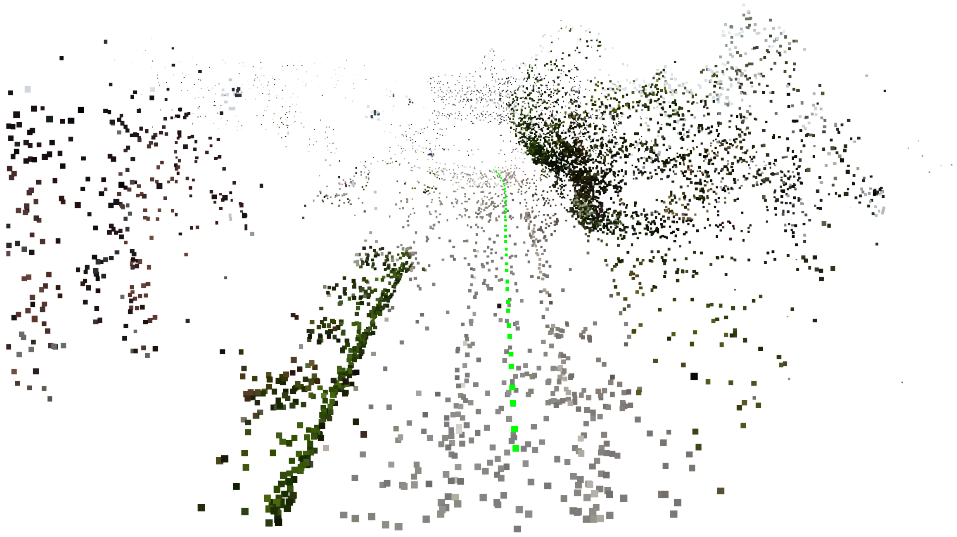


Figure 13. Example of a point cloud created from a vehicular video using SfM. The bright green points show the camera locations. Notice the sparsity of the cloud compared to the laser-scanned point cloud presented in Figure 9a).

3.5.1 Camera–road orientation

If the orientation of the camera and the road surface are known for a frame, the relative orientation between the camera and road surface is their orientation difference. Since the camera orientations are known, this leaves the task of estimating the road surface orientation. The point clouds that OpenMVG produces are relatively sparse. This is the case especially with the road surface, since usually the speed of the vehicle is high enough that motion blur occurs, leading to worse conditions for stable feature detection and matching.

The assumption that the camera is installed to a vehicle in a static way, provides a useful constraint which assists in solving the problem of determining the road surface orientation: While this constraint applies, the motion of the camera itself contains information about the road orientation: For example, if the camera points directly in the direction of the movement, we know that the road must be perfectly level relative to the camera from the x-axis perspective. Rather than using the sparse point cloud of the road surface to determine the road surface orientation, we can use the camera locations which have been calculated using the relative motions of the whole scene structure.

By analyzing the interaction between the camera orientation and movement, we can estimate the relative road orientation: For each frame used in the reconstruction,

a window of preceding and following frames is selected and the camera locations and rotations of these frames are stored. The windowed approach was used so that the effect possible errors present in single frames would be diminished. The window size used was 30 frames and was determined experimentally. For each of the stored location points, denoted by 3-vector $\mathbf{c}_i$, an additional point $\mathbf{p}_i$ is created using the rotation matrix of the camera $\mathbf{R}_i$ as follows:

$$\mathbf{p}_i = \mathbf{R}_i^\top \mathbf{v} + \mathbf{c}_i, \text{ where } \mathbf{v} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\top. \tag{8}$$

In Equation 8, $\mathbf{v}$ is the unit direction vector of side-ways motion relative to the camera. Using these camera locations and the additional points, we can form a plane by fitting it to these points in the least-squares sense using SVD. After the plane is formed, its normal vector can be used as an approximation of the normal vector of the road plane. The process is visualized in Figure 14. Using the retrieved normal vector, we can calculate the orientation difference between it and the camera. When we take into account the assumption of the negligibility of the camera's roll angle β, only the x-axis component is retrieved from $\mathbf{n}$. After the relative orientation difference has been calculated for each frame used in the reconstruction, a mean value is calculated and then used for creating the homography $\mathbf{H}_{\text{IPM}}$ which is then used throughout the video.

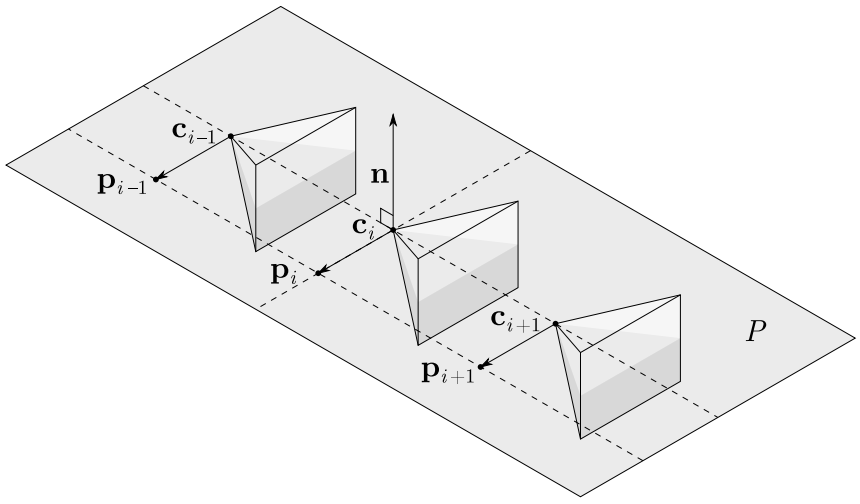


Figure 14. Formation of plane P for estimating the road orientation using camera centers and additional alignment points. The camera of the examined frame is denoted by $\mathbf{c}_i$. The preceding camera is $\mathbf{c}_{i-1}$ and the subsequent one is $\mathbf{c}_{i+1}$. The normal vector of the plane P is denoted by $\mathbf{n}$.

3.5.2 Camera heading deviation

The absolute heading γ for each frame in the GPS coordinate space is needed for the compositing of the orthoimages. The GPS data provided by an Android device includes a heading estimate for each data point, but their calculation is based on the movement of the camera. In cases where the heading of the camera does not match the vehicle's heading, the heading estimate cannot be directly used for the compositing. The heading difference between the camera and the vehicle $\Delta\gamma$ has to be first applied to the rotation matrix of the virtual camera as described in Section 2.3.

Since the 3D reconstruction provides us the orientations of the video frames and the reconstructions are georegistered, we can use that information to obtain $\Delta\gamma$. The absolute heading or the yaw angle of a single camera can be calculated as follows: First, we need the camera rotation matrix $\mathbf{R}_w$ in the world coordinate frame, the calculation of which is detailed in Section 4.3.1. The heading vector $\mathbf{v}_z$ of $\mathbf{R}_w$ is then calculated with $\mathbf{v}_z = \mathbf{R}_w^\top \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$. The absolute heading angle can then be determined by calculating the angle between $\mathbf{v}_z$ and a baseline vector defining the heading value 0. For each frame for which the pose estimation has been successful, we determine the difference between the heading obtained via reconstruction and GPS data. Then, a median value of the differences is calculated, defining $\Delta\gamma$.

3.5.3 Camera height

It is relatively simple to estimate the distance between the camera and the road surface (camera height) using the georegistered reconstruction of the scene: For each frame in the reconstruction, we select points from the point cloud for which the distance to the camera corresponding to the current frame is under 5 meters. This distance threshold was determined experimentally. From the selected points, we filter out those points which are above the camera. Then, using these remaining points we fit a plane using the least-squares method with SVD. Once the plane is defined, we calculate the orthogonal Euclidean distance between the plane and the camera. The final estimate of the camera height, which is used to calculate the IPM transform for all frames, is the mean value of all measurements from all the frames.

4 3D RECONSTRUCTION PRE- AND POST-PROCESSING

This section discusses processing methods which can be applied to the reconstruction data besides the actual reconstruction process. With 3D reconstruction being the foundation of the automatic camera calibration method of choice for the IPM transform, it is important to ensure that the reconstruction data is as robust as possible. The pre-processing methods presented in Sections 4.1 and 4.2 have proven themselves useful in increasing the successfulness and robustness of 3D reconstructions. The georegistration process of reconstructions is introduced in Section 4.3.

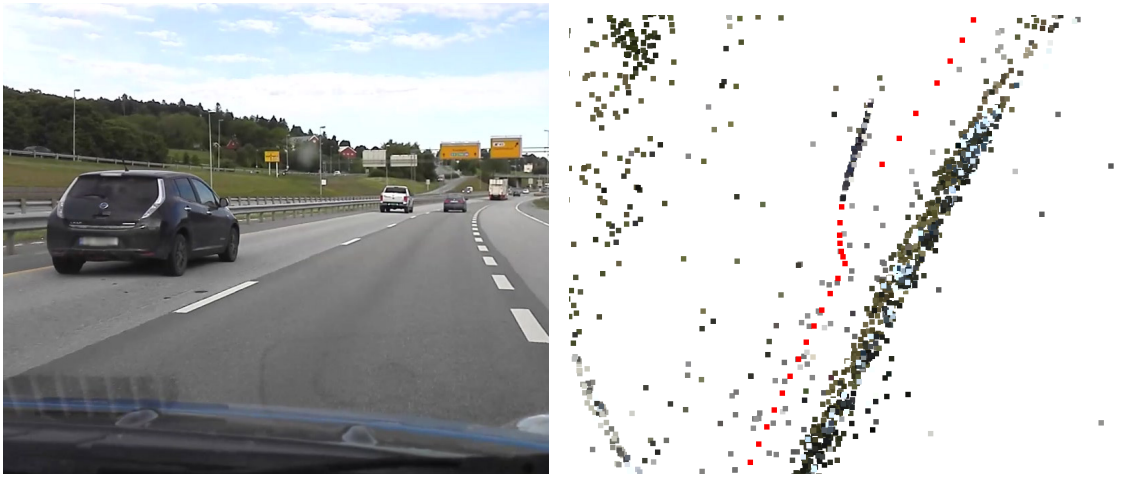
4.1 Sequence segmentation

The more images we use for a SfM reconstruction, the more data there is to optimize in the BA. The optimization cost of the objective function of BA is cubic in complexity, which is tied to the number of used images [42]. This limitation led to an approach where the reconstruction would be processed in separate segments to keep the processing times manageable. The fact that the reconstruction may fail at any point for various reasons also advocated the use of the segmentation approach. If a single segment fails, it is not as detrimental compared to a reconstruction of a whole video failing. For example, if a segment fails due to an anomaly in the images, we can attempt the reconstruction again, but start after the failed section.

4.2 Frame masking

One of the key assumptions which need to hold to ensure a successful 3D reconstruction is the staticity of the scene. Dynamic scenes, where there are other moving objects besides the camera, can result in inaccurate estimation of the camera poses and the global structure. The presence of moving objects, such as vehicles and people, is unavoidable in the context of road scenes. A method for reducing the deteriorating effects of moving objects was required and a simple way to accomplish this was frame masking: Before the calculation of the feature descriptors is carried out for the images, moving objects are masked so that no feature descriptors will be created for them. Objects which have no associated feature descriptors are effectively non-existing in the context of SfM.

During the development of the reconstruction system, it was observed that particularly other moving vehicles in the scene had a substantial effect on the accuracy of the camera poses and the global structure estimation. An example of this effect can be seen in Figure 15. In the scenario, the surveying vehicle moved straight ahead, but an overtaking car caused significant errors in the estimation of the camera poses. When the other moving vehicles are masked out, these kinds of errors are greatly reduced.



a) b)

Figure 15. Effects of a dynamic scene in 3D reconstruction: a) Video frame with an overtaking vehicle; b) Top view of the reconstruction. The red points denote the camera positions. Notice how the overtaking vehicle affects the trajectory of the recording vehicle, which is linear in reality.

The automatic masking of vehicles required their automatic detection, which was accomplished using the method proposed by Viola and Jones [43], which is based on the cascade classification of Haar-like features. The specific implementation which was used can be found in the OpenCV software library [44]. The robustness and accuracy of the method are not exactly state-of-the-art, but it produced adequate results for the frame masking and the computational performance of the method is high while using only a Central Processing Unit (CPU).

The road images often contain large areas which contain no relevant information for the reconstruction, namely the sky. By masking out the sky, we can slightly speed up the feature matching process. The sky may also cause errors in the reconstruction if clouds are present: The clouds can be so distant that they appear static in the image sequence while the scene close to the camera is clearly moving. The detection of sky pixels was achieved with an algorithm based on detecting large contours in the image

with a high average pixel intensity which are also connected to the upper boundaries of the image frame. The algorithm utilizes image thresholding and morphological operations for forming the contours.

The vehicle hood can cause errors in the reconstruction since it conflicts with the motion estimation: When the surrounding scene moves relative to the camera, the hood stays in place. With smaller vehicles, it is often difficult to install the camera to the windshield so that the vehicle's hood would not be visible in the frame. An algorithm was created for the automatic masking of parts of the image which are static throughout a video. The approach was based on absolute image differences for frame pairs selected randomly from the video. The image difference results would be then averaged out. After about 150 image differences, an image would form where the static parts of the video would have lower intensity values compared to the rest of the image. Often the road surface in front of the camera is relatively static when it comes to pixel intensity changes, which has caused problems.

4.3 Georegistration

When a 3D reconstruction is created using SfM with no prior data on the absolute locations of the images, the coordinate frame of the result does not match the real world. The relative structure is obtained, but meaningful absolute distances cannot be measured, for example. Through the process of georegistration, a 3D reconstruction can be tied to the real world by transforming it to a world coordinate frame, enabling further applications for the data.

Let us assume we have two sets of 3D points, set $\mathbf{S}_s$ and $\mathbf{S}_t$. The source point set $\mathbf{S}_s$ contains the camera locations in the coordinate frame which the reconstruction process has defined. The target point set $\mathbf{S}_t$ contains the corresponding camera locations in the world coordinate frame. By finding the 4×4 affine transformation matrix $\mathbf{T}$ which minimizes the difference between the transformed source point set $\mathbf{TS}_s$ and the target point set $\mathbf{S}_t$, we can georegister the entire reconstruction, both camera poses and the point cloud. The transformation matrix $\mathbf{T}$ includes translation, rotation, and uniform scaling components.

The set $\mathbf{S}_t$ can be defined using the collected GPS coordinates, which leaves the task of determining set $\mathbf{S}_s$. The GPS receiver sampling rate and frame times are not related to each other. If we interpolate the camera positions in the reconstruction

frame to match the sample times of the GPS points, we obtain the corresponding source point set $\mathbf{S}_s$. The GPS points are initially expressed in World Geodetic System (WGS) coordinates. In order to find the optimal transformation, the GPS points need to be converted to Euclidean coordinates. The Universal Transverse Mercator (UTM) coordinate system was used for this case.

4.3.1 Affine transformation

The optimal transformation matrix $\mathbf{T}$ between the source point set $\mathbf{S}_s$ and the target point set $\mathbf{S}_t$ can be found using the method proposed by Kabsch [45], which is based on the minimization of weighted sums of squared deviations. Besides the reconstruction, the GPS camera locations can have errors as well. For example, high rise buildings can have a deteriorating effect on the accuracy of the GPS readings [46]. Consequently, two methods for estimating the optimal transform robustly were applied, to reduce the effect of possible outliers in the source and target point sets.

For each image sequence segment, a transformation model was computed using both methods and the model having the lowest mean error would be selected. The first method was RANdom SAmple Consensus (RANSAC) [47]. The second was an iterative method, where in each iteration, point pairs with an error exceeding a threshold t were removed from the model as described in Algorithm 2. The main loop will be exited if the maximum number of iterations is reached, no point pairs were deleted, or the number of point pairs reaches the minimum of 3. The threshold t was defined as $t = \bar{e} + 3\sigma_e$, where $\bar{e}$ is the mean and σ_e the standard deviation (normal distribution) of the model error. The error of the model for a single point pair was defined as the Euclidean distance between the source point, which has been transformed by the model, and the corresponding target point.

Transforming points, such as the camera locations or cloud points from a coordinate frame to another is a simple task: With the transformation matrix $\mathbf{T}$, a point in homogeneous coordinates $\mathbf{x}$ can be transformed to $\mathbf{x}'$ with the matrix multiplication $\mathbf{x}' = \mathbf{T}\mathbf{x}$. Transforming the camera rotations to a new coordinate frame is not so straightforward. There are at least two approaches for solving the problem. One possibility is the decomposition of the transformation matrix $\mathbf{T}$ to its translation, rotation, and scaling components by methods proposed by Thomas [48] and Goldman [49]. We can then apply the rotation component $\mathbf{R}_t$ to a camera rotation $\mathbf{R}_r$ in

Algorithm 2: Robust iterative affine transformation estimation algorithm

input : $\mathbf{S}_s$ - source point set
$\mathbf{S}_t$ - target point set
m - maximum number of iterations
output: $\mathbf{T}$ - affine transformation matrix
for $i = 1, \dots, m$ **do**
  Compute model $\mathbf{T}$ with sets $\mathbf{S}_s$ and $\mathbf{S}_t$
  Calculate mean $\bar{e}$ and standard deviation σ_e error of the model $\mathbf{T}$
  Calculate error threshold $t = \bar{e} + 3\sigma_e$
  $d = 0$
  for $j = |\mathbf{S}_s|, \dots, 1$ **do**
    if $\|\mathbf{T}\mathbf{S}_s[j] - \mathbf{S}_t[j]\|_2 > t$ **then**
      Delete $\mathbf{S}_s[j]$ and $\mathbf{S}_t[j]$
      $d = d + 1$
  if $d = 0 \vee |\mathbf{S}_s| = 3$ **then**
    Break from loop

the reconstruction frame, to obtain the rotation in the world frame $\mathbf{R}_w$ as follows:

$$\mathbf{R}_w = \mathbf{R}_t \mathbf{R}_r. \tag{9}$$

Alternatively, transforming camera rotations to a new coordinate frame without the decomposition of the transformation matrix $\mathbf{T}$ can be accomplished as follows: The camera rotation $\mathbf{R}_w$ in the world frame can be thought to be composed of four column unit 4-vectors:

$$\mathbf{R}_w = \begin{bmatrix} \mathbf{x}_w & \mathbf{y}_w & \mathbf{z}_w & \mathbf{w} \end{bmatrix} \tag{10}$$

where the vectors $\mathbf{x}_w$, $\mathbf{y}_w$, and $\mathbf{z}_w$ define the axes of the world camera frame and $\mathbf{w} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^\top$. The world camera frame, reconstruction camera frame, their corresponding axes, and the georegistration process are visualized in Figure 16.

To ensure that the axes of the world camera frame are of unit length, let us define these vectors as follows by using $\mathbf{x}_w$ as an example:

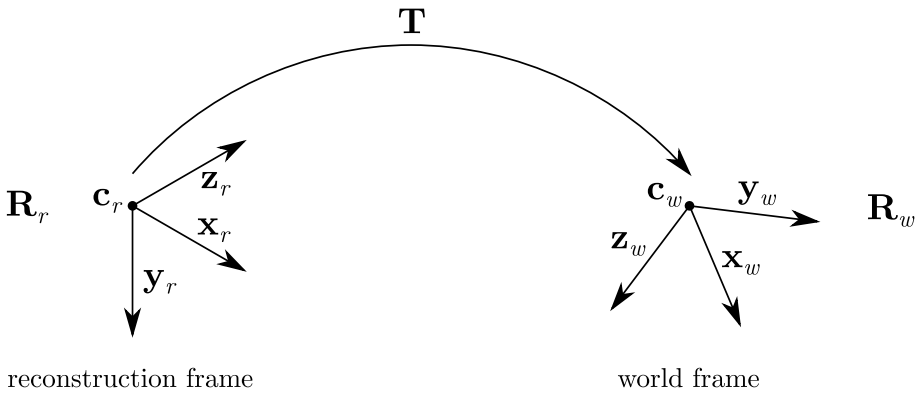


Figure 16. Camera georegistration.

$$\mathbf{x}_w = \frac{\mathbf{v}_x}{\|\mathbf{v}_x\|_2}. \tag{11}$$

To define these vectors $\mathbf{v}_*$, we require their start and end points. The starting point $\mathbf{c}_w$ is the camera location in the world coordinate frame and the ending point $\mathbf{p}_w$ is the corresponding point $\mathbf{p}_r$ in the reconstruction frame transformed by $\mathbf{T}$. Both points are in homogeneous coordinates. For example, the vector $\mathbf{v}_x$ can be expressed as follows:

$$\mathbf{v}_x = \mathbf{p}_w - \mathbf{c}_w = \mathbf{T}\mathbf{p}_r - \mathbf{c}_w. \tag{12}$$

The camera position $\mathbf{c}_w$ can also be expressed using $\mathbf{T}$ and so the previous expression takes the form $\mathbf{v}_x = \mathbf{T}\mathbf{p}_r - \mathbf{T}\mathbf{c}_r = \mathbf{T}(\mathbf{p}_r - \mathbf{c}_r)$. The point $\mathbf{p}_r$ is defined as:

$$\mathbf{p}_r = \mathbf{c}_r + \mathbf{x}_r = \mathbf{c}_r + \mathbf{R}_r^\top \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\top. \tag{13}$$

In Equation 13, $\mathbf{x}_r$ is the vector which defines the x-axis in the reconstruction camera frame, which corresponds to vector $\mathbf{x}_w$ in the world camera frame. Using this definition we can further the expression of $\mathbf{v}_x$:

$$\begin{aligned} \mathbf{v}_x = \mathbf{T}(\mathbf{p}_r - \mathbf{c}_r) &= \mathbf{T}\left(\mathbf{c}_r + \mathbf{R}_r^\top \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\top - \mathbf{c}_r\right) \\ &= \mathbf{T}\left(\mathbf{R}_r^\top \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\top\right). \end{aligned} \tag{14}$$

Now that we have the definition of vector $\mathbf{x}_w$, we can apply it to vectors $\mathbf{y}_w$ and $\mathbf{z}_w$ as well. As a whole, the formation of the camera world rotation $\mathbf{R}_w$ can be calculated as follows:

$$\begin{aligned}
\mathbf{R}_w &= \begin{bmatrix} \mathbf{x}_w & \mathbf{y}_w & \mathbf{z}_w & \mathbf{w} \end{bmatrix} \\
\mathbf{x}_w &= \frac{\mathbf{v}_x}{\|\mathbf{v}_x\|_2} \quad \text{and} \quad \mathbf{v}_x = \mathbf{T}\mathbf{R}_r^\top \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\top \\
\mathbf{y}_w &= \frac{\mathbf{v}_y}{\|\mathbf{v}_y\|_2} \quad \text{and} \quad \mathbf{v}_y = \mathbf{T}\mathbf{R}_r^\top \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}^\top \\
\mathbf{z}_w &= \frac{\mathbf{v}_z}{\|\mathbf{v}_z\|_2} \quad \text{and} \quad \mathbf{v}_z = \mathbf{T}\mathbf{R}_r^\top \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^\top \\
\mathbf{w} &= \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^\top .
\end{aligned} \tag{15}$$

4.3.2 Gravity alignment of linear segments

In the georegistration process, a problem can occur when the movement of the camera is linear, i.e., the vehicle has moved in a straight line: The linearity of the source and target camera points introduce an additional degree of freedom in finding the optimal transformation. The reconstruction camera positions can rotate by any amount around the axis defined by the movement direction. In this case, additional information is needed on the orientation of the cameras in the world coordinate frame in order to find an accurate transformation $\mathbf{T}$.

When the recording vehicle moves straight forward at a constant speed, forces which affect the smartphone are small. Here, the accelerometer readings of the smartphone provide a reasonable estimate for the gravitational acceleration. Using the tri-axis data, the orientation of the camera can be approximated. The accelerometer sampling rate does not match the frame times. With linear interpolation, a gravity vector can be determined for each frame. Using the gravity vectors $\mathbf{g}_i$, we can create additional points $\mathbf{p}_i$ to the source point set $\mathbf{S}_s$ as follows:

$$\mathbf{p}_i = \mathbf{c}_i + \frac{\mathbf{R}_i^\top \mathbf{v}_i}{s}, \text{ where } \mathbf{v}_i = \frac{\mathbf{g}_i}{\|\mathbf{g}_i\|_2}. \tag{16}$$

In Equation 16, $\mathbf{c}_i$ denotes the camera position and $\mathbf{R}_i$ the camera rotation in the original coordinate frame. The scalar s is the uniform scale factor in the transforma-

tion matrix $\mathbf{T}$. Corresponding to these additional points in $\mathbf{S}_s$, we also add points to the target set $\mathbf{S}_t$ as follows:

$$\mathbf{p}_i = \mathbf{c}_i + \mathbf{v}, \text{ where } \mathbf{v} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^\top. \tag{17}$$

In Equation 17, $\mathbf{c}_i$ denotes the camera position in the UTM coordinate system and $\mathbf{v}$ is the unit vector pointing up in the vertical direction in the coordinate system. After the additional points have been added to sets $\mathbf{S}_s$ and $\mathbf{S}_t$, the optimal transformation $\mathbf{T}$ is again found, with the extra degree of freedom removed.

When the camera is not moving in a linear fashion, the accelerometer readings most likely do not reflect the camera orientation accurately. Thus, the gravity alignment is utilized only for segments where the camera has moved in a straight line. These kinds of segments are automatically detected by fitting a line using the camera position points $\mathbf{c}_i$. If the orthogonal distances between of all of the points and the line do not exceed a threshold, the segment is classified as linear. The line fitting is done in 2D, with the altitude component omitted.

4.3.3 Altitude filtering

The altitude readings provided by a typical Android device's GPS sensor are much more inaccurate compared to the longitude and latitude readings. The altitude values have also been discretized to a step of 1 meter. The altitude data is filtered in order to have it reflect the real world more accurately. The filtering method used was Gaussian kernel convolution. The kernel is defined as

$$K(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right), \tag{18}$$

where σ is the standard deviation [50]. The boundary effects in the kernel convolution are handled by using only those parts of the kernel for which corresponding data points are available. The kernel is always normalized, whether the kernel is used fully or not. An example of the GPS altitude data convolution filtering is presented in Figure 17.

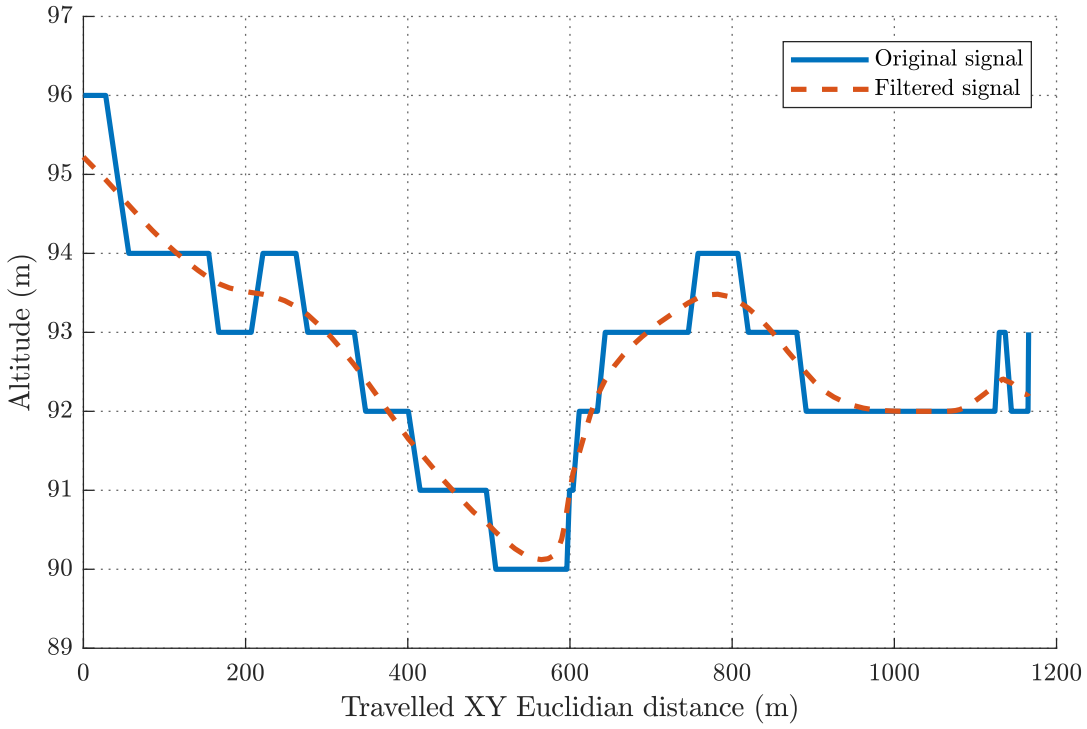


Figure 17. Gaussian kernel convolution of GPS altitude data, where the kernel size $w = 21$ and the standard deviation $\sigma = 4.0$.

5 SYSTEM IMPLEMENTATION

This section outlines the implementation of the components which finalize the orthoimagery generation system, including the creation of orthophoto composites and the extraction of road surface features. Factors which can affect the quality of the orthoimages are discussed and solutions are provided for the problem cases. Lastly, the complete system is recapitulated and viewed as a whole.

5.1 Orthophoto composite generation

The orthoimages created from individual video frames are not that useful on their own. By using the collected GPS data and the absolute camera headings, the IPM frames can be composited together, creating a map layer of the road surface which can cover a large distance. The GPS locations are interpolated for each frame since the sampling rates of the GPS receiver and the camera do not match. In order to position the images accordingly, a pixel position $\mathbf{p}_i$ and a heading angle γ_i is needed for each IPM image.

The calculation of the pixel positions requires the georegistration of the individual IPM images. In this case, the georegistration of an IPM image consists of determining the UTM coordinates of the top-left and bottom-right corners of the IPM image, which has been rotated by the angle γ_i, the absolute heading of the frame. The georegistration begins with the calculation of the image scale ratio ω, which denotes the number of pixels in the distance of one meter.

The calculation of ω is carried out as follows: First, we calculate where the physical camera is located on the image plane of the virtual camera. The projection matrix $\mathbf{P}$ is calculated for the virtual camera and the position of the physical camera $\mathbf{x}'$ is determined with the projection $\mathbf{x}' = \mathbf{P}\mathbf{x}$, where $\mathbf{x}$ is the physical camera's 3D location. After $\mathbf{x}'$ is retrieved, we can measure the distance d_q between $\mathbf{x}'$ and the center of the IPM image $\mathbf{q}$ in pixels. Since the forward translation t_z between the physical and virtual camera is known in world coordinates, ω can be calculated using the pixel and world coordinate distance with $\omega = d_q/t_z$.

The UTM coordinates of the IPM image center can be calculated using t_z. Now that we have two points in the IPM image for which we know the UTM coordinates, we can retrieve the UTM coordinates of the IPM image corners. After the IPM

images have been georegistered, the image pixel positions $\mathbf{p}_i$ can be calculated with $\mathbf{p}_i = \omega \mathbf{u}_i$, where $\mathbf{u}_i$ is the UTM coordinate of the top-left corner of the IPM image. The georegistration process is visualized in Figure 18.

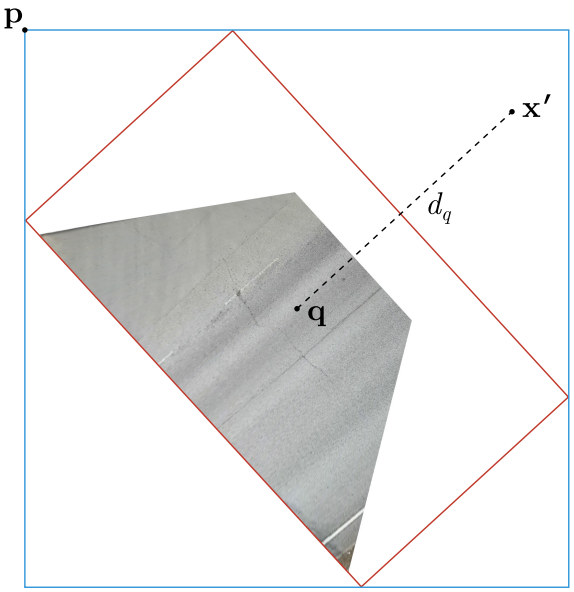


Figure 18. IPM image georegistration. The red frame denotes the bounds of the IPM image before rotation and the blue frame after. The position of the physical camera on the virtual camera's image plane is denoted by $\mathbf{x}'$. The center of the IPM frame is denoted by $\mathbf{q}$. The top-left corner of the rotated IPM frame is denoted by $\mathbf{p}$.

The IPM images are composited in chronological order. This way, the most accurate part of the IPM images is preserved in the composite, given that the vehicle has been moving forward. The last IPM image added in the composite can be seen fully as it is not obstructed by other images. The georegistration of the composite can be achieved by simply calculating the bounding rectangle for all the used images. Figure 19 presents a comparison between Google satellite imagery and a orthoimage composite generated with the system.

When the vehicle is stationary during data recording, the GPS signal often starts to deteriorate, which leads to erroneous orthophoto composites. Solving the problem requires using the other available sensors to determine if the vehicle is stationary. Using SfM to determine this is not a reliable solution, as the reconstruction can fail for reasons other than the absence of camera movement. One possibility to determine the vehicle immobility could be the usage of the accelerometer. The solution should take into account the factor that the engine of the vehicle can cause strong periodic vibrations which can make the solving of the problem less straightforward. Figure 20 presents frequency spectra of tri-axis accelerometer data for trucks and smaller cars.

a) b)

Figure 19. Comparison of orthoimage resolution: a) Google satellite image of a highway; b) Previous image with a generated orthophoto layer superimposed.

The sampling rate of the accelerometer is 500 Hz meaning that the Nyquist frequency is 250 Hz. From the figure we can see that a common primary frequency cannot be determined and most of the spectra contain multiple peaks. The strongest vibrations occur below 80 Hz.

5.2 Road surface features

The orthophoto composites can be augmented by applying the IPM transform to annotation data in the original frames. This annotation data can consist of pixel masks which identify certain objects in the scene. As a result, we receive an augmented composite which includes the annotation data. For example, if the road markings and other road surface features have been annotated, the results can be projected and composited to an orthophoto. The annotation of objects of interest was accomplished with semantic segmentation, a method which pursues to understand images at the pixel level: Each pixel in an image is given a label corresponding to the class of the object the pixel is part of [51, 52]. The specific implementation of semantic segmentation used was Full-Resolution Residual Networks (FRRN) [53] proposed by Pohlen et al. which has been designed specifically street scenes in mind.

When semantic segmentation is applied to an image, the resulting segmentation image contains the class information for each pixel. The class information is encoded in the image using the pixel gray-value. Each class has its own unique gray-value.

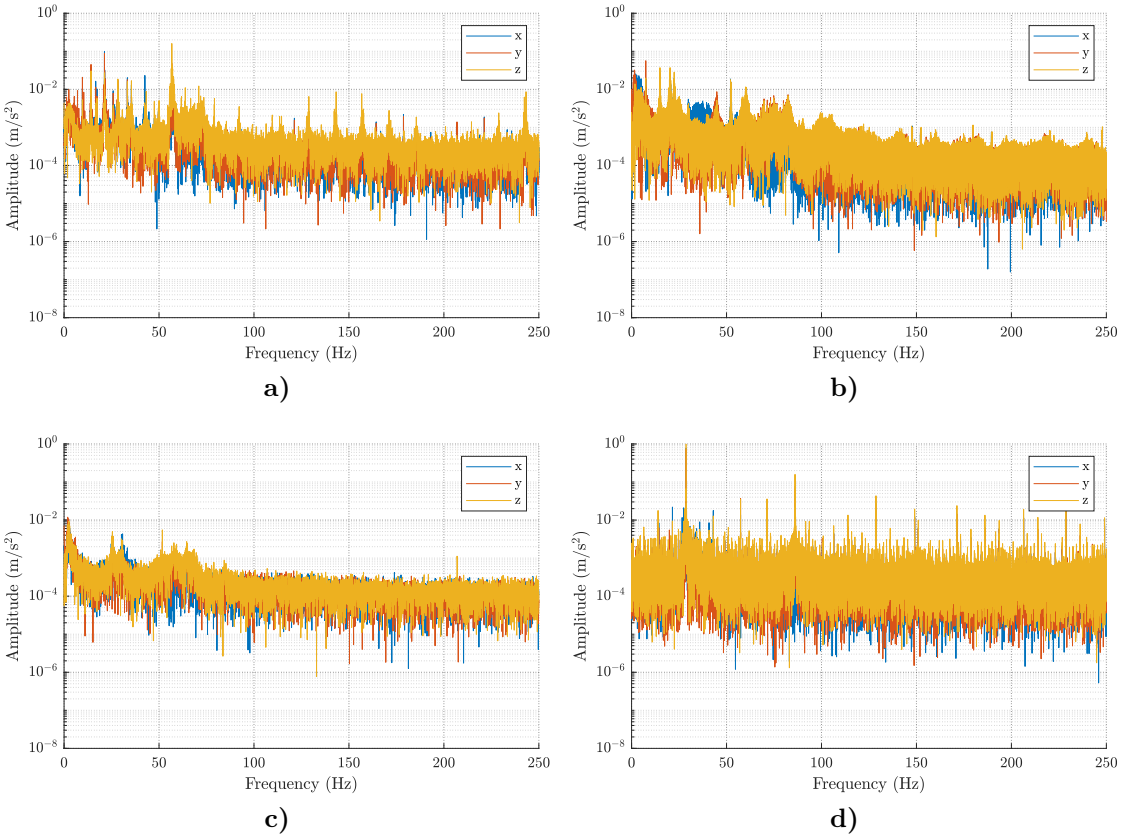
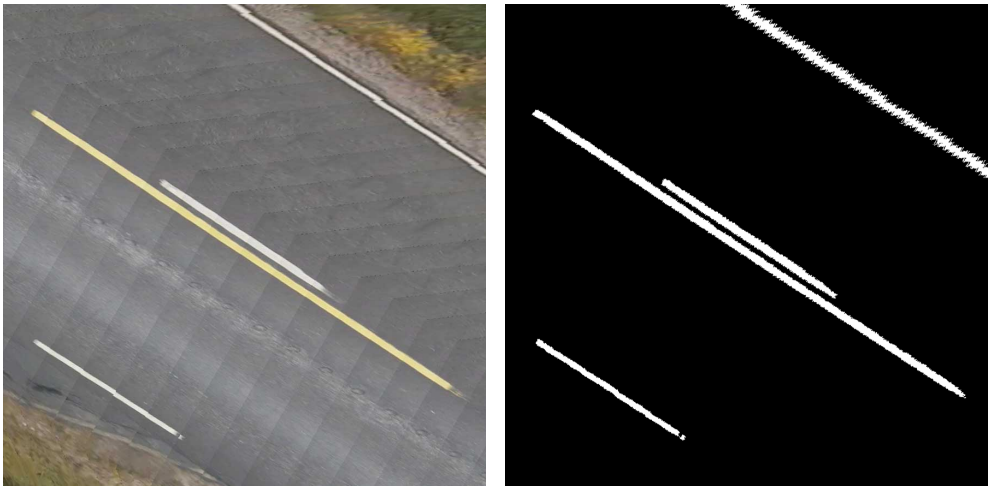


Figure 20. Frequency spectra for tri-axis accelerometer data. The data collection vehicles were stationary with the engine on. Subfigures a) and b) show data from trucks. Subfigures c) and d) show data from smaller cars. Note that the vertical axes are logarithmic.

Unlike in the case of the original frames, when applying the IPM transform to the segmentation images, nearest-neighbor interpolation must be used, because linear interpolation can introduce errors to the result: Assume that we have two neighboring areas segmented to different classes. If linear interpolation were to be used in IPM, in the resulting IPM frame the boundary of the two areas would contain gray-values that would not present either of the two classes. When nearest-neighbor interpolation is used, the IPM frame will have the same gray-values as the original.

From the augmented composite, various road features, such as lane markings, road cracks, and potholes, can be extracted by masking the composite with the appropriate gray-value of the class in question. After the masking, the contours presenting the road features can be extracted from the binary image. An example feature mask can be seen in Figure 21.

The condition evaluation of road markings can be carried out as follows: A histogram is computed for the road surface using its segmentation mask. The histogram is



a) b)

Figure 21. Extraction of road features: a) Orthoimage composite; b) Corresponding segmentation orthoimage with the road markings mask visible.

computed from the value channel after the image is converted to the Hue Saturation Value (HSV) color space. Similarly, a histogram is computed for each of the road marking contours. All of the road marking histograms are then compared to the road histogram using the Bhattacharyya distance [54] as a similarity measure. Markings similar to the road are classified as being in poor condition and vice versa.

5.3 Camera positioning

The manner how the camera is installed to the vehicle plays an integral part in the final orthophoto accuracy. When applying the IPM to the original image in a typical road scene, pixels at the bottom of the image are closer to each other compared to their initial state. After a certain point, the pixels begin to be farther from each other as we proceed upwards in the image. When comparing these distances of sequential pixels in the original and warped image, the relative differences can be used as multipliers for the base pixel size. After applying the multipliers to the base pixel size, we can survey how the inverse-perspective image quality degrades as we move upwards in the image.

Figure 22 presents two different scenarios for camera installation. The vehicle used in the first scenario is a truck and in the second it is a smaller passenger car. The scenarios are presented in Subfigures a) and b) respectively and include the original and IPM frames. In Subfigure c), the effective pixel size of the inverse-perspective

image is shown as a function of the relative vertical position of the original image for both scenarios. The effective pixel sizes were calculated for the IPM images using the base pixel size ω^{-1}. In both of the scenarios, the original image resolution was 1920×1080.

In the truck scenario, the hood of the vehicle is practically vertical. This means that the bottom part of the frame can be utilized in its entirety, which is the most accurate part of the inverse-perspective frame. The distance between the camera and the road was 2.1 meters and the camera–road orientation θ was 2.9 degrees. In the car scenario, the hood of the vehicle is elongated. The car hood covers a large part of the bottom of the image, rendering that part unusable for the inverse-perspective frame. With most passenger cars, this problem cannot be completely avoided. After the road becomes visible in the inverse-perspective frame, the quality of the image has already degraded greatly. The distance between the camera and the road was 1.5 meters and the camera–road orientation θ was 0.7 degrees.

In the truck scenario, the effective pixel size of the lowest usable position is 8.2 mm and in the passenger car case it is 14.2 mm. Even though the camera is higher from the road in the truck scenario, the effective pixel size is smaller in the usable image region compared to the car scenario. From these observations, we can reason that using vehicles with shorter hoods is preferable for creating road orthophotos.

5.4 Image obstructions

The video frames can contain many features which can impair the quality of the resulting orthoimage. These effects can be bad purely from a human perspective, i.e., the result may not be pleasing to look at. In some cases, subsequent computer vision tasks can also be negatively affected. These kinds of features or image obstructions can be divided into two categories: static and dynamic. Next, these two types of obstructions and their countermeasures are explored.

5.4.1 Static

As detailed in Section 5.3, the vehicle hood can significantly decrease the quality of the orthophotos, since less of the accurate area of the original image remains. Before the image compositing can be done, the pixels which belong to the car hood

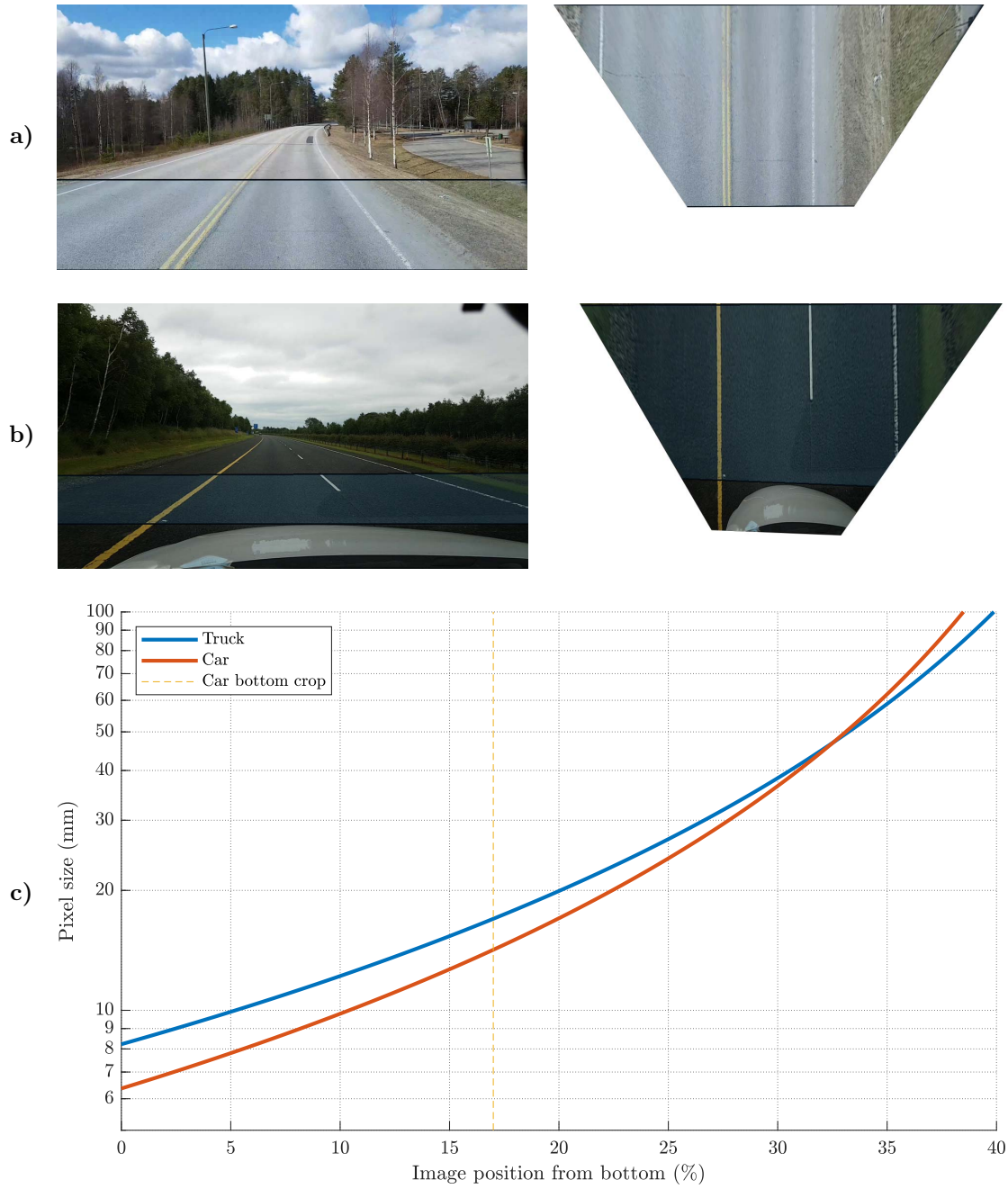


Figure 22. Effect of camera positioning on orthophoto accuracy. Subfigure a) shows the original and IPM frames for the truck scenario. Subfigure b) shows the original and IPM frames for the car scenario. A blue overlay shows the usable part of the image in both subfigures. Subfigure c) presents the effective pixel size in the IPM frame as the function of the vertical position of the original image. Position 0% is the bottom of the image. The dashed yellow line shows the location of the bottom of the usable image area in the car scenario. Note that the vertical axis is logarithmic.

need to be automatically detected, to prevent the hood from ending up in the final composite. An example of what occurs when the hood is not masked out from the images before compositing can be seen in Figure 23. The problem can be solved by masking out static objects using semantic segmentation.

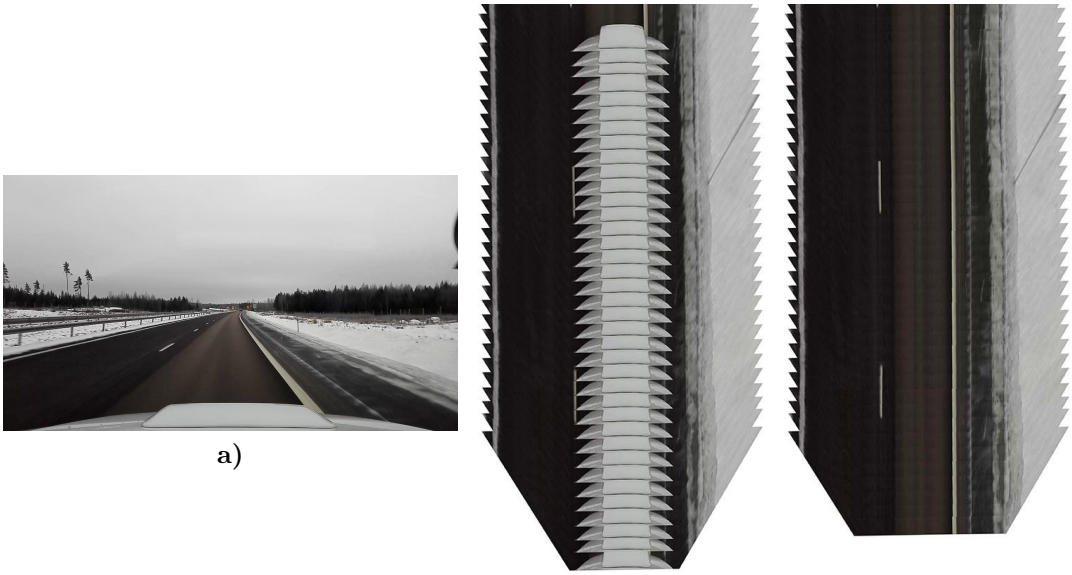


Figure 23. Effects of a visible car hood for an orthophoto composite: a) Original frame from a video; b) Image composite with a visible car hood; c) Image composite with the car hood masked.

Reflections caused by diffuse light in the vehicle's windshield can be problematic in the same sense as the vehicle hood. Although semantic segmentation is not affected by these kinds of reflections, the orthophoto composite may look unappealing to a human eye, as the reflections can create a repeating pattern. One possible hardware solution could be the installation of a polarizing lens in front of the camera, which could remove or at least attenuate reflections. This solution may have the drawback of increased motion blur in the video: Due to the lens, the amount of light reaching the camera is reduced, which may prompt the camera to lower the shutter speed automatically. Effects of a polarizing filter can be seen in Figure 24.

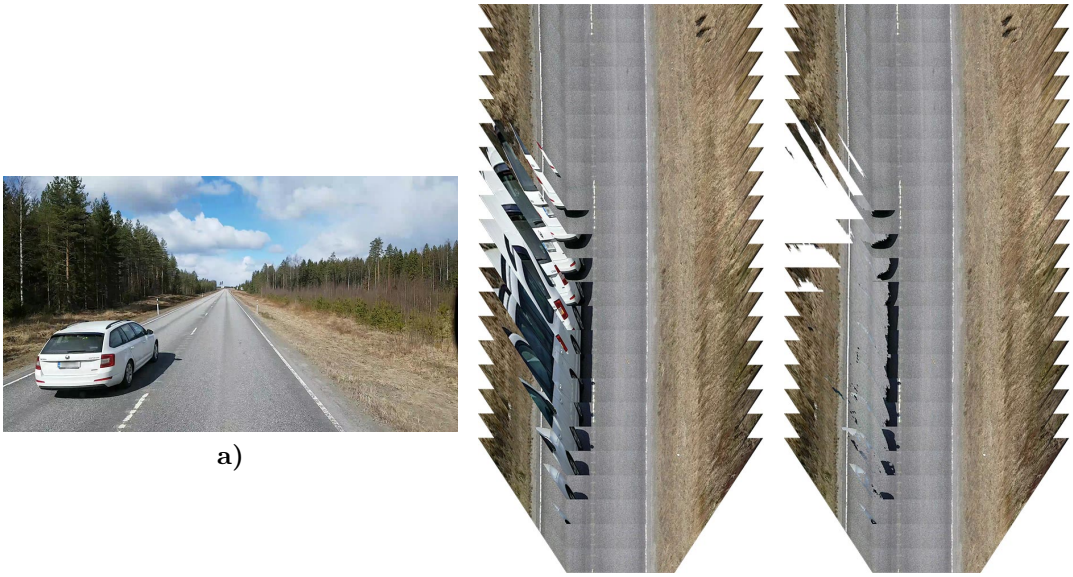
5.4.2 Dynamic

Other vehicles on the road are not objects of interest in the context of road surface orthophotos and they do not fulfill the planarity assumption. For these reasons,

a) b)

Figure 24. Effects of a polarizing lens in a dash cam scenario: a) Image without a polarizing lens with the reflection of the vehicle's dashboard (dark horizontal stripes) outlined; b) Image with an adjusted polarizing lens.

other vehicles should be masked out of the IPM frames. An overtaking vehicle can have a detrimental effect on numerous consecutive frames in the orthophoto, for example. Again, semantic segmentation is a good solution for masking out vehicles so they do not end up in the orthophoto composite. An example of masking out an overtaking vehicle from a composite can be seen in Figure 25.



a) b) c)

Figure 25. Masking of an overtaking vehicle in an orthophoto composite: a) Original frame from a video; b) Image composite with no masking; c) Image composite with masking enabled.

5.5 System overview

A complete overview of the orthoimage generation system is presented in Figure 26. The raw data sources are marked with gray and the two major subsystems (camera calibration and orthophoto generation) with dashed lines. The images provided by the camera are used in the camera calibration, semantic segmentation, and orthophoto generation. The GPS coordinates are used in the camera calibration and the georegistration of the orthophoto composites. The accelerometer readings are used only in the camera calibration.

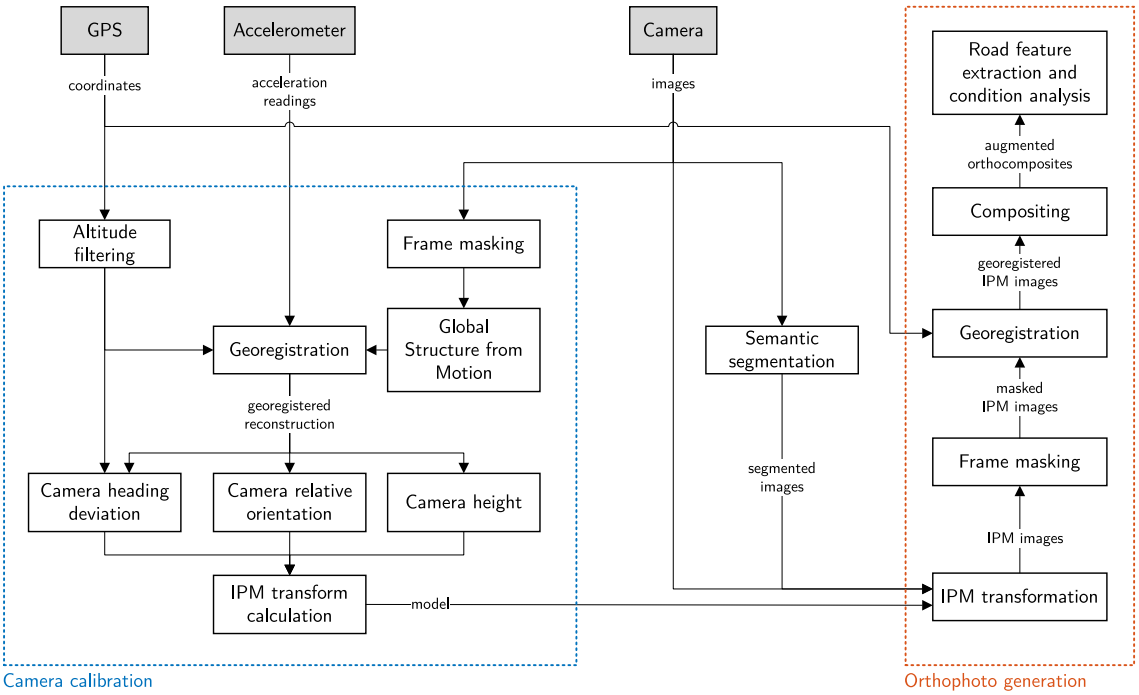


Figure 26. Overview of the orthoimage generation system.

In the camera calibration, the masked camera frames are used to create a 3D reconstruction with global SfM and the results are then georegistered using filtered GPS and raw accelerometer data. The necessary components for the calculation of the IPM transform are derived from the reconstruction. The retrieved IPM transformation is then used on the original and semantically segmented camera frames, from which unwanted objects are subsequently masked. The IPM frames are then composited and georegistered using GPS and heading data. After composition, the road surface features can be extracted using the semantic segmentation data.

6 EXPERIMENTS AND RESULTS

This section presents the experiments which were carried out to test the accuracy of the chosen camera calibration method. The general approach of the experiments is described and then the experiment results are presented for the core components of the solution.

6.1 Approach and dataset

The SfM based camera calibration method presented in Section 3.5 was experimented on regarding its accuracy. The three core components examined were the following:

- The orientation difference between the camera and the road surface about the x-axis (pitch), denoted by θ. It will be referred as the camera–road orientation from now on since it is its sole considered component.
- The orientation difference between the camera and the vehicle about the y-axis (yaw), denoted by $\Delta\gamma$.
- The height of the camera from the road surface, denoted by d_p.

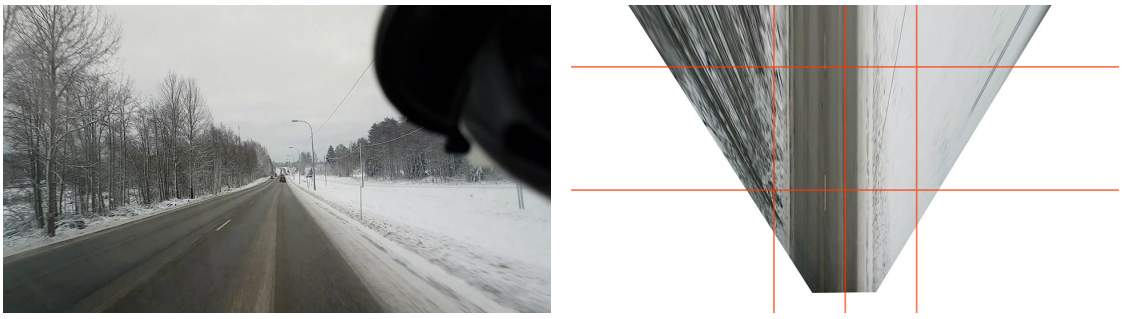
All of the variables are visualized in Figure 7. With all of the components, a single test case consisted of calculating an estimate for the variable in question and then calculating the absolute error regarding the ground truth value.

Even though Vionice possesses a vast and diverse database of road videos, it could not be used for assembling a dataset for testing. This is because the camera height d_p is not known beforehand for any of the videos and cannot be measured reliably afterward. A new dataset for the testing of all components was recorded by the author on the field. Sample frames from the dataset can be seen in Figure 27. The true camera heights were physically measured before the data collection.

The ground truth values for θ and γ were determined by hand after the data collection, by testing what values would produce an accurate IPM transform: For each video, segments with a straight road were selected and values for θ and γ were tuned to produce a result where the edges of the road would be parallel and straight. A grid was used to help determine if the IPM transform was accurate which can be seen in Figure 28. The virtual camera was positioned higher than normal during this process, increasing the length of the visible road segment in the IPM frame. This

Figure 27. Frames from the dataset demonstrating the different conditions and environments present.

adjustment enabled more accurate parameter tuning, as errors in the configuration become more apparent as we move farther from the camera in the IPM frame.



a) **b)**

Figure 28. Determining ground truth values for θ and γ: a) Original frame from a video; b) IPM frame with an alignment grid.

The dataset included multiple different camera installations with varying orientations and heights. Most of the videos in the dataset had a length of five minutes, which is the upper limit with the recording software. In total, there were 3 hours and 26 minutes of footage. All of the videos had a resolution of 1920×1080 and a frame rate of 30 frames per second. To increase the sample size for the experimentation on each component, the videos were divided into roughly four parts. By increasing

the sample size, the way the errors are distributed becomes more apparent.

The dataset was collected during two non-consecutive days and includes scenes with road, rural, suburban, and urban environments. Normal and snowy weather conditions can be found in the dataset. Examples for each environment and condition class are presented in Figure 29. The definition of snowy conditions was not based on the requirement that there should be snow on the road surface, but if there is any snow present in the environment. The different environment and condition classes were taken into account in the experiments to discover how they are related to the observed estimation errors.

The average speed of the recording vehicle was taken into account in the experiments with all of the components. When the speed of the recording vehicle is high, the accuracy of the 3D reconstruction may be negatively affected due to motion blur, which can make the extraction of stable features more difficult. Since all of the experiment components depend on the 3D reconstructions, it is worthwhile to test if a quantitative correlation can be measured between the estimation errors and the corresponding vehicle speeds. The average accuracy of the GPS readings was considered in the experiments with the camera height estimation component. The accuracy of a GPS location is given in the Android operating system as a meter reading, which is the radius of a circle. The center of the circle is defined by the GPS coordinate. By the definition, the true device location is inside the circle with a 68% confidence [55].

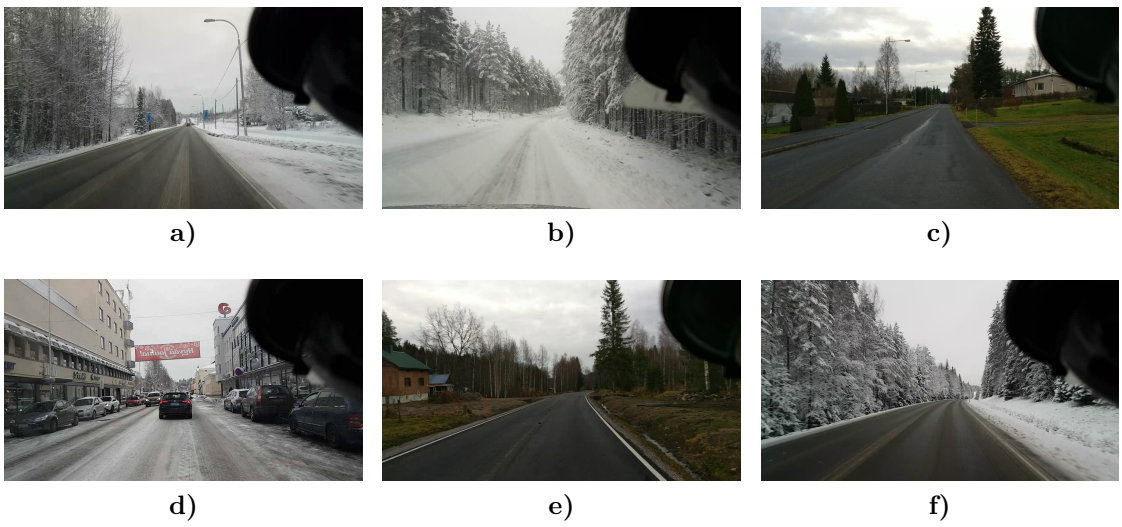


Figure 29. Examples of the environment and condition classes. Environment classes: a) Road; b) Rural road; c) Suburban; d) Urban. Condition classes: e) Normal; f) Snow.

6.2 Parameters

The working resolution used with the Haar cascade classifier was 1440×810 for shorter processing times. The working resolution in SfM was 1920×1080 (original image resolution). The used feature descriptor was Scale-Invariant Feature Transform (SIFT) [56], which is the default feature descriptor in OpenMVG version 1.2. The used descriptor parameters can be seen in Table A1.1, which were determined experimentally. The video mode matching was used with the SfM feature matching, where consecutive frames are matched to each other. The other SfM parameters used were the defaults in OpenMVG version 1.2. For more information about the default parameters, access the OpenMVG documentation [57]. The experimentally determined parameters used in reconstruction georegistration can be found in Table A1.2.

6.3 Experiment results

With the estimation of the camera–road orientation θ, the absolute mean error was 1.22 degrees and the median was 0.84 degrees. The distribution of the absolute errors is presented in Figure 30. If we divide the errors by environment class, we can see that the mean error was significantly higher in the *road* class compared to the others, as shown in Figure 31a). Regarding the condition classes, the *normal* class shows a much higher mean error compared to the *snow* class, as shown in Figure 31b).

The absolute mean error was 0.97 degrees and the median was 0.58 degrees with the estimation of the camera heading deviation $\Delta\gamma$. The distribution of the absolute errors is presented in Figure 32. Similar to the camera–road orientation estimation, the *road* class had the highest mean error among the other environment classes, as shown in Figure 33a). Figure 33b) shows that the *normal* condition class had a much larger mean error compared to the *snow* class, as was the case with the camera–road orientation estimation.

In the case of estimation of the camera height d_p, the absolute mean error was 9.77 cm and the median was 7.15 cm. Figure 34 shows the distribution of the absolute errors. Figure 35a) shows that the observed errors were the highest in samples belonging to the *rural road* condition class. The prevalent weather conditions seemed to have minimal effect on the errors as Figure 35b) shows. Figure 36a) shows that

the GPS accuracy was fairly equal in all environments except in the *urban* class where it was slightly worse, which is excepted as large buildings can interfere with the GPS signal.

The threshold lines in Figures 30, 32, and 34 show the acceptable maximum error for each of the experiment components. The definition of these error thresholds is introduced in Section 7.2.1.

A scatter plot of absolute camera height estimation errors and corresponding average GPS accuracy values is shown in Figure 36b). Figure 37 shows scatter plots of the absolute experiment errors and the corresponding average vehicle speeds. For the previously mentioned cases, the intervariable correlation coefficients are presented in Tables 2 and 3 using the Pearson correlation coefficient and the Spearman rank correlation coefficient, respectively. The corresponding p-values are also presented in these tables.

The Pearson correlation coefficient measures linear correlation between two variables [58]. The Spearman correlation coefficient is calculated by applying the Pearson correlation to the ranks of the observations in the data [59]. In other words, the coefficient assesses how well a monotonous function can describe the relationship between two variables. The Spearman correlation was taken into consideration as well because we have no prior indication that the relationships between the variables would be linear in nature.

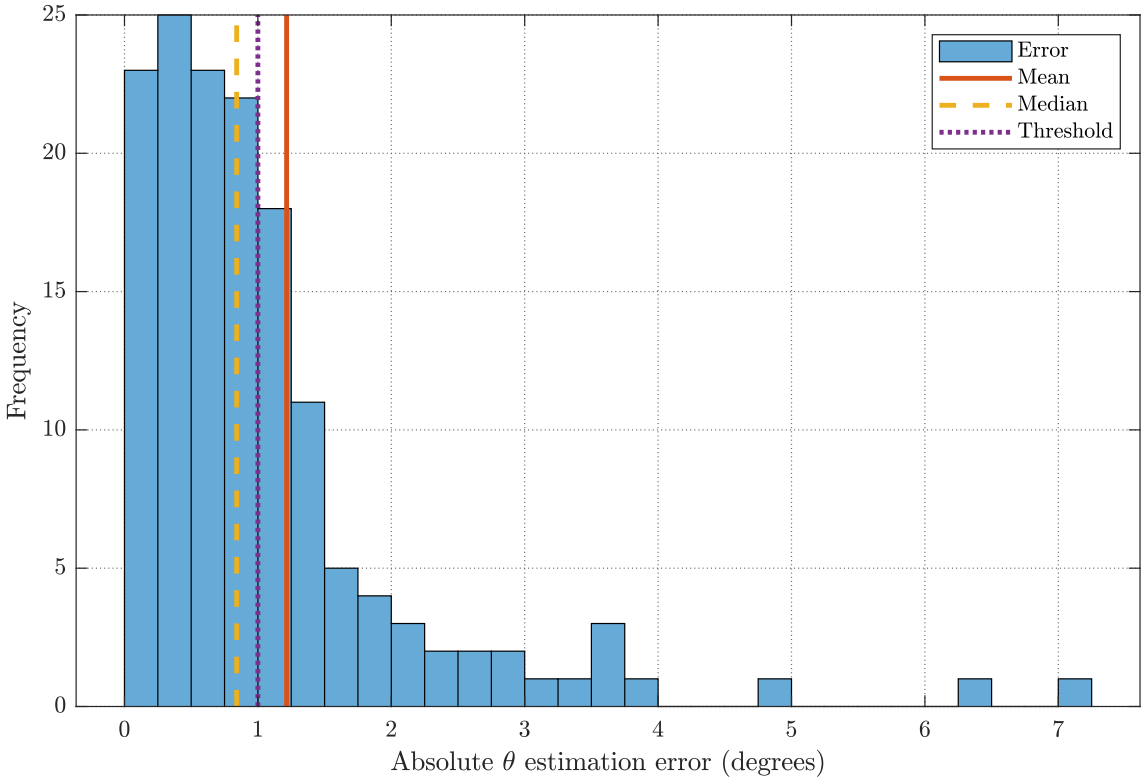


Figure 30. Histogram of absolute camera–road orientation estimation errors. For the visual clarity of the histogram, two samples with the absolute errors of 10.75 and 13.68 degrees are not displayed. Both of these samples belong to the *road* environment class.

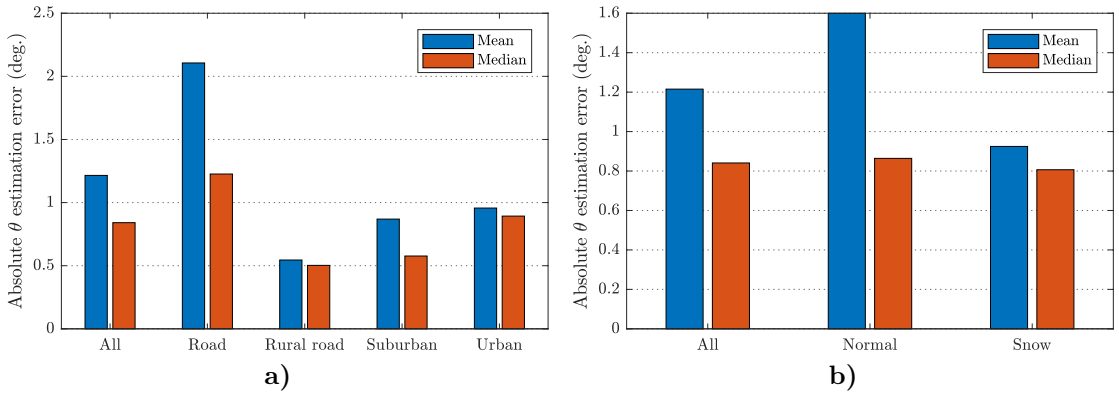


Figure 31. Bar plots of absolute camera–road orientation estimation errors: a) Errors grouped by environment classes; b) Errors grouped by condition classes.

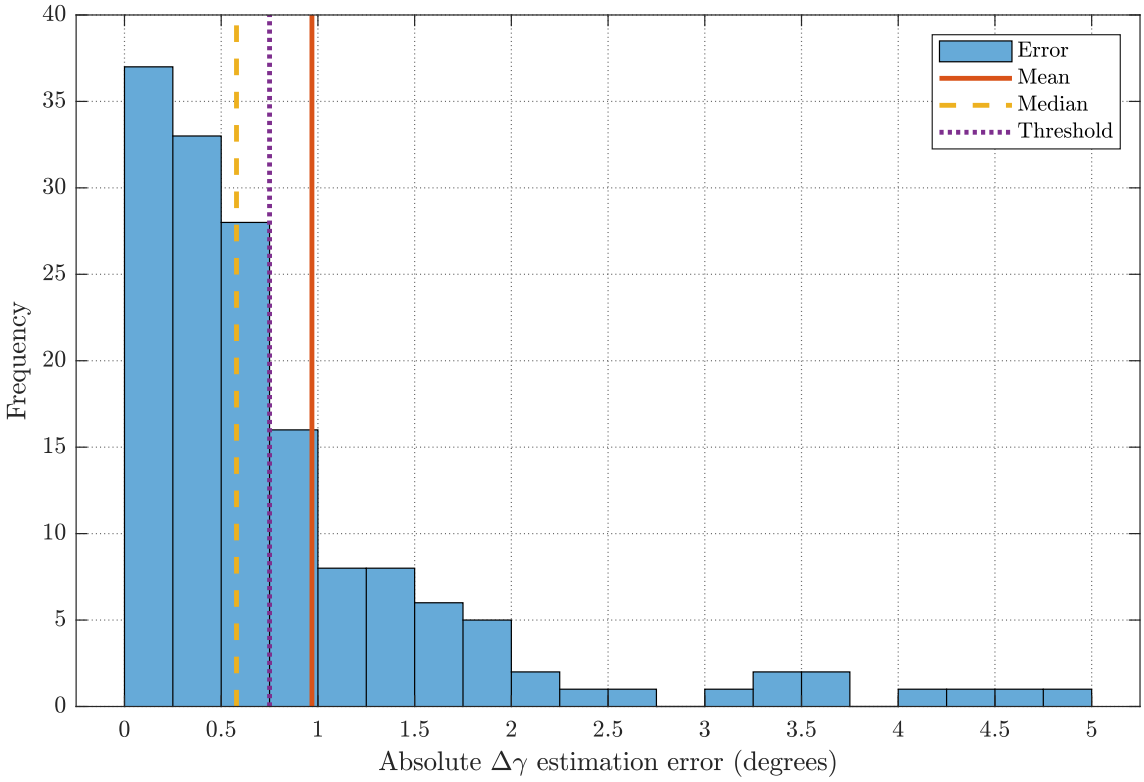


Figure 32. Histogram of absolute camera heading deviation estimation errors. For the visual clarity of the histogram, two samples with the absolute errors of 8.75 and 11.12 degrees are not displayed. Both of these samples belong to the *road* environment class.

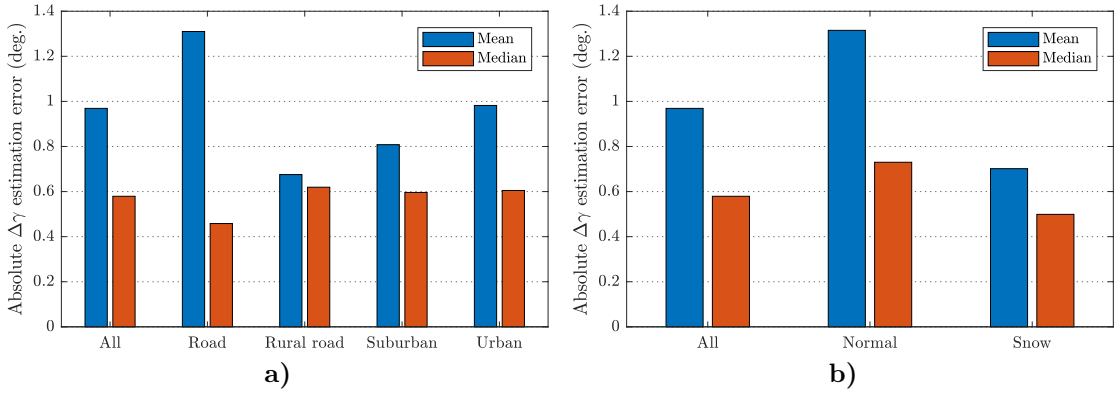


Figure 33. Bar plots of absolute camera heading deviation estimation errors: a) Errors grouped by environment classes; b) Errors grouped by condition classes.

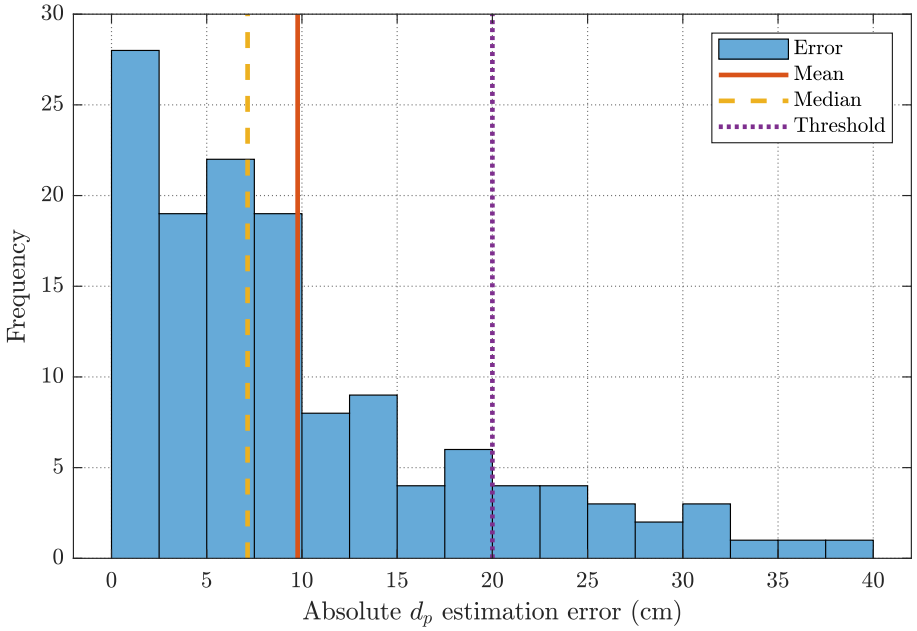


Figure 34. Histogram of absolute camera height estimation errors.

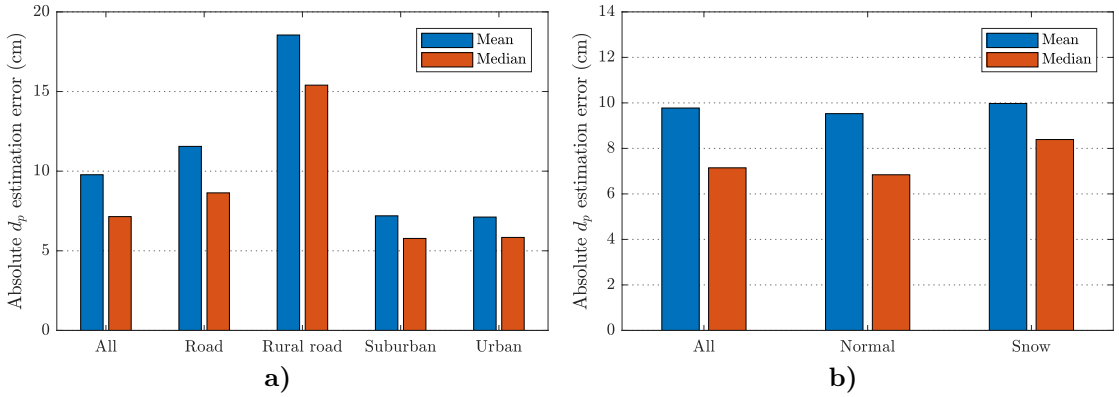


Figure 35. Bar plots of absolute camera height estimation errors: a) Errors grouped by environment classes; b) Errors grouped by condition classes.

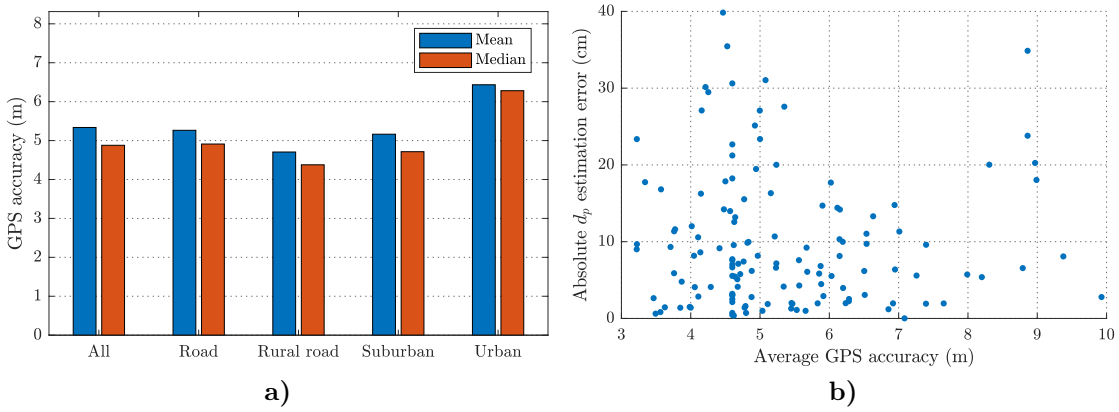


Figure 36. Figures on GPS accuracy: a) GPS accuracies grouped by environment classes; b) Scatter plot of absolute camera height estimation errors and corresponding average GPS accuracy values.

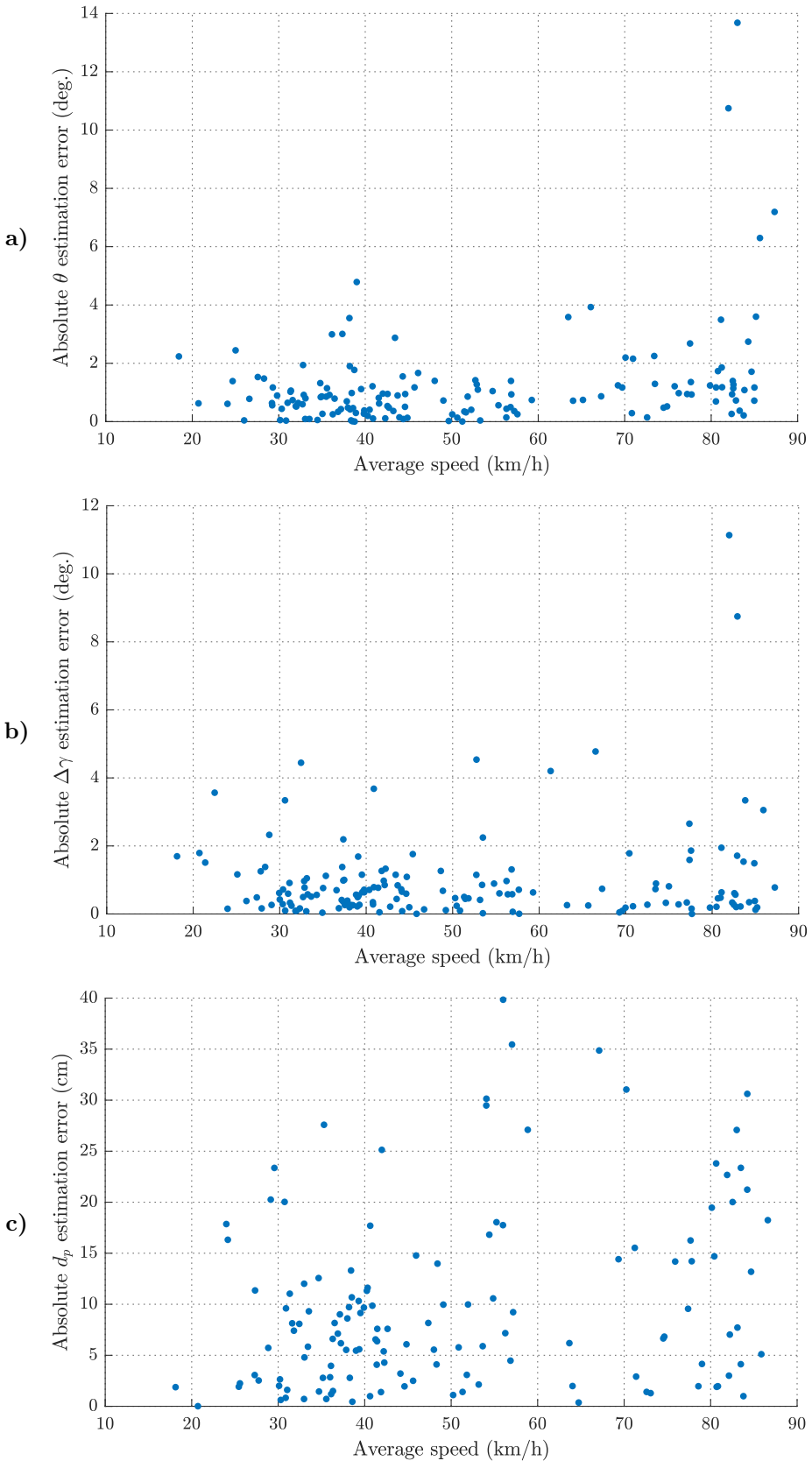


Figure 37. Scatter plots of the absolute experiment errors and the corresponding average vehicle speeds: a) Camera–road orientation estimation; b) Camera heading deviation estimation; c) Camera height estimation.

Table 2. Correlations between the experiment variables using the Pearson correlation coefficient.

Variable 1	Variable 2	Coefficient	p-value
θ abs. estimation error	Average vehicle speed	0.3211	5.5829×10^{-5}
$\Delta\gamma$ abs. estimation error	Average vehicle speed	0.1357	0.0912
d_p abs. estimation error	Average vehicle speed	0.2719	0.0015
d_p abs. estimation error	Average GPS accuracy	−0.002	0.9813

Table 3. Correlations between the experiment variables using the Spearman rank correlation coefficient.

Variable 1	Variable 2	Coefficient	p-value
θ abs. estimation error	Average vehicle speed	0.2472	0.0023
$\Delta\gamma$ abs. estimation error	Average vehicle speed	−0.02	0.8026
d_p abs. estimation error	Average vehicle speed	0.246	0.0042
d_p abs. estimation error	Average GPS accuracy	−0.04	0.6441

7 DISCUSSION

This section analyses the results attained in Section 6 to resolve if the performance of the system implementation is adequate and how different factors affect the results. Possibilities for the further development of the system are also discussed.

7.1 Intervariable correlations

The absolute camera height d_p estimation error and the average GPS accuracy show no clear visible correlation, as can be seen in Figure 36b). The same applies to the average vehicle speed and all of the experiment components, as shown in Figure 37. Next, the quantitative correlation measures shown in Tables 2 and 3 are reviewed. A significance level of 0.01 will be used when assessing the p-values in the tables.

By reviewing the correlation tables, we can see that the absolute estimation errors of the camera–road orientation θ and the camera height d_p show a weak positive correlation regarding the average vehicle speed. The correlation coefficients are in the range of 0.246–0.3211. The p-values are 0.0042 or lower, signifying that the probability of observing this kind of correlation is highly unlikely in the case that the variables were uncorrelated in reality. Since the p-values are well below the used significance level, the correlations are regarded as statistically significant.

The other intervariable relationships (absolute camera heading deviation $\Delta\gamma$ estimation error and average vehicle speed, absolute camera height d_p estimation error and average GPS accuracy) show no evidence of correlation, as the correlation coefficients are in the range of -0.04–0.1357. If the variables are not correlated in reality, the received low correlation coefficients are likely, because the p-values are in the range of 0.0912–0.9813.

7.2 System performance analysis

An understanding of how different errors affect the orthophoto results is required in order to survey the performance of the system implementation. Next, a maximum acceptable error threshold is determined for each of the experiment components by using the visual quality of the orthoimages as the main criterion.

7.2.1 Error thresholds

How the estimation error of the camera–road orientation θ deteriorates the end result can be seen in Figure A2.1. By viewing the figure, we can reason that an error between 0.0 and 1.0 degrees does not significantly impair the visual quality of the orthophoto composite. The visual artifacts can be seen clearly in the composite when the error reaches 1.5 degrees. Based on these observations, an error of 1.0 degrees is regarded as the acceptable maximum.

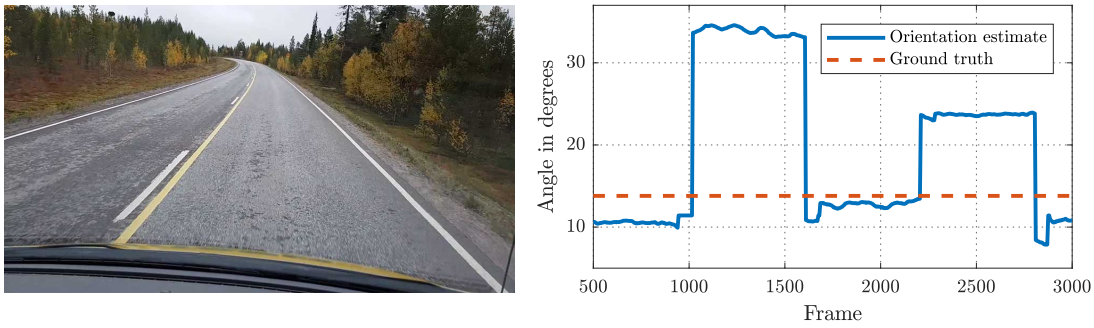
Figure A2.2 demonstrates how the camera heading deviation $\Delta\gamma$ estimation error affects the quality of orthophoto composites. When the error is between 0.0 and 0.75 degrees, the visual quality of the composite is not substantially compromised. As the error reaches 1.0 degrees, the visual artifacts seem too prevalent. The acceptable maximum error for the component is set at 0.75 degrees.

The effects of the camera height d_p estimation error are visualized in Figure A2.3. We can see that an error between 0 and 20 cm does not seem to reduce the quality of the orthophoto composite meaningfully. When the error reaches 30 cm, the orthophoto appears to be too distorted. Thus, the acceptable maximum error for the camera height estimation is set at 20 cm in the evaluation of the robustness of the method.

7.2.2 Analysis of results

Looking at the results presented in Section 6.3, we can say that the accuracy of the camera–road orientation estimation is not adequate, as 38% of the samples exceeded the acceptable maximum error threshold of 1.0 degrees. A probable cause for the errors could be the instability of image features which can lead to errors in the rotations estimation. Empirical tests have shown that depending on the conditions in the video, the estimates for the camera–road orientation can fluctuate wildly between segments. An extreme case of this is presented in Figure 38, where a large portion of the image is covered by the road.

The road surface often has little or no texture, leading to fewer feature points and less robust feature matching. When the data collection vehicle is moving at high speed, the road surface near the camera can suffer from motion blur, making its contribution to the previously mentioned problem. In Figure 38a), the only trackable regions are



a) b)

Figure 38. Highly varying camera–road orientation estimates within a single video: a) Single frame from the video; b) Graph showing the ground truth and the changing orientation estimate. The different segments can be identified by the locally even sections.

the vegetation on both sides of the road, which is a rather small part of the whole image, making the camera pose estimation more difficult since fewer keypoints are available.

Figure 39 shows the same road scene with the SIFT features plotted. The scales of the keypoints extracted from the road surface are relatively small on average, making them more unstable from the tracking viewpoint. The keypoints for which a match was not found in the subsequent frame are marked with red. For most of the keypoints belonging to the road surface, a match was not found. The obtained results support these observations since samples belonging to the *road* class had the highest mean error in the camera–road orientation estimation.

37% of the samples exceeded the acceptable maximum error of 0.75 degrees in the camera heading deviation estimation experiments, indicating its performance as insufficient. The component is dependent almost solely on the 3D reconstructions, like the camera–road orientation component. The previously mentioned problems regarding the reconstructions concern the camera heading deviation component as well.

The estimation of camera–road orientation and camera heading deviation both demonstrated a much higher mean error with the *normal* environment class compared to the *snow* class. This may be due to issues experienced regarding the automatic focus of the camera, which was used during the day the videos which belong to the *normal* class were recorded. Occasionally, the recorded frames were not fully in focus, deteriorating the reconstruction results. As both of the components are highly dependent on the reconstructions, their performance degraded expectedly.

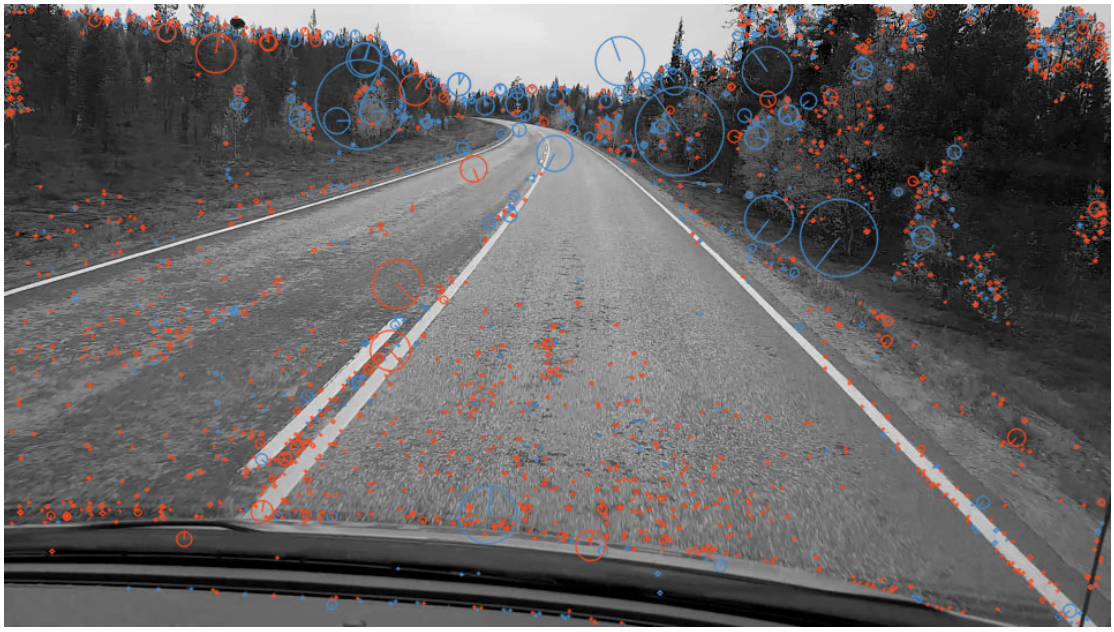


Figure 39. Visualization of SIFT keypoints. The keypoints for which there is a match in the subsequent frame are marked with blue and the keypoints for which a match has not been found are marked with red.

In the case of the accuracy of camera height estimation, 14% of the samples exceeded the error threshold of 20 cm, which is a much better result than with the previous components. The *rural road* environment class had the highest mean and median error. With the camera–road orientation estimation, the same class showed a low mean and median error compared to the other classes, implying successful pose estimation, which again suggests successful structure estimation. In the likely case that the accuracy of the GPS data has no influence on the camera height estimation error, it is possible that the errors in the *rural road* class are caused by inaccurate structure estimation of the road surface. These inaccuracies may be caused by the instability of feature points extracted from the road surface, as was previously discussed with the camera–road orientation estimation errors within *road* class.

7.3 Future work

One possible approach for improving the estimation accuracy of the camera–road orientation and camera heading deviation could be the tuning of the feature descriptor parameters for SfM or using a different feature descriptor altogether. If this proves to be ineffective, another possibility is to replace the camera pose estimation method (SfM) completely. The latter option may be necessary, because it seems unlikely that drastic errors for example seen in Figure 38b) could be corrected by

just parameter tuning. The methods based on optical flow, which were presented in Section 3.4, could be good candidates for an alternative solution.

The camera height estimation accuracy could be improved by employing RANSAC in the plane formations and applying constraints to the plane formation e.g., the steepness of the plane should not exceed a certain limit. Usually multiple videos are recorded with the same camera installation in succession, i.e., videos often belong to a session of videos. Another viable approach for improving the performance of all the components of the camera calibration would be the expansion of the estimation context from a single video to all of the videos in a session. The more data there is available in the camera calibration, the more accurate results are potentially produced.

Apart from improving the performance of the camera calibration, future development could include the implementation of the following:

- Employing semantic segmentation in the reconstruction frame masking process, completely replacing the initial algorithms.
- Detecting if the data collection vehicle is stationary automatically as described in the end of Section 5.1.
- Extraction and analysis of more useful road feature data from the orthophoto composites augmented by semantic segmentation, e.g., road width.

8 CONCLUSION

The objective of this study was to create a robust system for generating road orthoimages using only a smartphone installed to a vehicle. The creation of the orthoimages was based on IPM. The primary research problem was set to be the calibration of the camera for the IPM transform. An introduction to the theory behind the IPM transform was given. Different methods for camera calibration for IPM and orthophoto generation in general were surveyed. A novel camera calibration method based on SfM reconstructions was chosen for the system implementation. Points on how 3D reconstruction results could be improved were outlined, which may prove useful in other applications as well. In addition, the georegistration process for 3D reconstructions was introduced.

The implementation for the compositing and georegistration of orthophotos was explained. Factors which affect the quality of the orthophoto composite, e.g., camera positioning and image obstructions, were gone through and solutions were presented for problem cases. The extraction and condition analysis of road features were introduced, which were based on semantic segmentation and histogram similarity measures, respectively.

The proposed novel method for the camera calibration was experimented on regarding its robustness and accuracy using a road video dataset collected by the author. The experiment results demonstrated that the accuracy of the calibration method was insufficient: Components which depend on the estimation of the relative orientation of the camera were not performing adequately as 38% of the samples exceeded the threshold of acceptable error on average. The camera calibration method requires further development to enable the orthophoto generation system to be put into use.

References

[1] R. Brunauer and K. Rehrl, "Supporting road maintenance with in-vehicle data: Results from a field trial on road surface condition monitoring," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2016, pp. 2236–2241.

[2] I. Sikirić, K. Brkić, and S. Šegvić, "Recovering a comprehensive road appearance mosaic from video," in *MIPRO, 2010 Proceedings of the 33rd International Convention*, IEEE, 2010, pp. 755–759.

[3] P. Palašek, P. Bosilj, and S. Šegvić, "Detecting and recognizing centerlines as parabolic sections of the steerable filter response," in *MIPRO, 2011 Proceedings of the 34th International Convention*, IEEE, 2011, pp. 903–908.

[4] J.-i. Meguro, H. Ishida, K. Kidono, and Y. Kojima, "Road ortho-image generation based on accurate vehicle trajectory estimation by GPS Doppler," in *Intelligent Vehicles Symposium (IV)*, IEEE, Jun. 2012, pp. 276–281.

[5] M. Yang, C. Wang, F. Chen, B. Wang, and H. Li, "A new approach to high-accuracy road orthophoto mapping based on wavelet transform," *International Journal of Computational Intelligence Systems*, vol. 4, no. 6, pp. 1367–1374, 2011.

[6] Z. Zhang, "Perspective camera," in *Computer Vision: A Reference Guide*, K. Ikeuchi, Ed. Boston, MA: Springer US, 2014, pp. 590–592. [Online]. Available: `https://doi.org/10.1007/978-0-387-31439-6_114`.

[7] P. Sturm, "Pinhole camera model," in *Computer Vision: A Reference Guide*, K. Ikeuchi, Ed., Springer US, 2014, pp. 610–613. [Online]. Available: `http://dx.doi.org/10.1007/978-0-387-31439-6_472`.

[8] Z. Zhang, "Camera parameters (intrinsic, extrinsic)," in *Computer Vision: A Reference Guide*, K. Ikeuchi, Ed., Boston, MA: Springer US, 2014, pp. 81–85. [Online]. Available: `http://dx.doi.org/10.1007/978-0-387-31439-6_152`.

[9] T. Nöll, A. Pagani, and D. Stricker, "Markerless camera pose estimation - an overview," in *Visualization of Large and Unstructured Data Sets - Applications in Geospatial Planning, Modeling and Engineering (IRTG 1131 Workshop)*, vol. 19, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2011, pp. 45–54. [Online]. Available: `http://drops.dagstuhl.de/opus/volltexte/2011/3096`.

[10] M. Bertozzi, A. Broggi, and A. Fascioli, "Stereo inverse perspective mapping: Theory and applications," *Image and Vision Computing*, vol. 16, no. 8, pp. 585–590, 1998. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S0262885697000930`.

[11] S. Tan, J. Dale, A. Anderson, and A. Johnston, "Inverse perspective mapping and optic flow: A calibration method and a quantitative analysis," *Image and Vision Computing*, vol. 24, no. 2, pp. 153–165, 2006.

[12] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[13] D. Yagishita and H. Chikatsu, "Generation of effective orthophotos for road surfaces using MMS," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 5, p. 621, 2014.

[14] B. Vallet and J.-P. Papelard, "Road orthophoto/DTM generation from mobile laser scanning.," *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2015.

[15] V. De Silva, J. Roche, and A. Kondoz, "Fusion of LiDAR and camera sensor data for environment sensing in driverless vehicles," *Computing Research Repository*, Oct. 2017. [Online]. Available: `http://arxiv.org/abs/1710.06230`.

[16] K. Kwak, D. F. Huber, H. Badino, and T. Kanade, "Extrinsic calibration of a single line scanning LIDAR and a camera," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 3283–3289.

[17] Y. Park, S. Yun, C. S. Won, K. Cho, K. Um, and S. Sim, "Calibration between color camera and 3D LIDAR instruments with a polygonal planar board," *Sensors*, vol. 14, no. 3, pp. 5333–5353, 2014.

[18] H. Oliveira and P. L. Correia, "Automatic road crack segmentation using entropy and image dynamic thresholding," in *17th European Signal Processing Conference*, IEEE, 2009, pp. 622–626.

[19] A. Kheyrollahi and T. P. Breckon, "Automatic real-time road marking recognition using a feature driven approach," *Machine Vision and Applications*, vol. 23, no. 1, pp. 123–133, 2012.

[20] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[21] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 679–698, 1986.

[22] M. Nieto, L. Salgado, F. Jaureguizar, and J. Cabrera, "Stabilization of inverse perspective mapping images based on robust vanishing point estimation," in *Intelligent Vehicles Symposium, 2007 IEEE*, IEEE, 2007, pp. 315–320.

[23] M. Fang, G. Yue, and Q. Yu, "The study on an application of Otsu method in Canny operator," in *International Symposium on Information Processing (ISIP)*, 2009, pp. 109–112.

[24] Y.-K. Huo, G. Wei, Y.-D. Zhang, and L.-N. Wu, "An adaptive threshold for the Canny operator of edge detection," in *2010 International Conference on Image Analysis and Signal Processing (IASP)*, IEEE, 2010, pp. 371–374.

[25] W. Rong, Z. Li, W. Zhang, and L. Sun, "An improved Canny edge detection algorithm," in *2014 IEEE International Conference on Mechatronics and Automation (ICMA)*, IEEE, 2014, pp. 577–582.

[26] T. Brox, "Optical flow," in *Computer Vision: A Reference Guide*, K. Ikeuchi, Ed. Boston, MA: Springer US, 2014, pp. 564–564. [Online]. Available: `http://dx.doi.org/10.1007/978-0-387-31439-6_100214`.

[27] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," Springer Berlin Heidelberg, 2003, pp. 363–370.

[28] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[29] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision – ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I*, A. Leonardis, H. Bischof, and A. Pinz, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 430–443.

[30] L. J. Rapo, "Feature-based camera pose estimation," Bachelor's thesis, Lappeenranta University of Technology, Jun. 2016.

[31] K. Schindler and W. Förstner, "Photogrammetry," in *Computer Vision: A Reference Guide*, K. Ikeuchi, Ed. Boston, MA: Springer US, 2014, pp. 597–599. [Online]. Available: `https://doi.org/10.1007/978-0-387-31439-6_139`.

[32] M. Westoby, J. Brasington, N. Glasser, M. Hambrey, and J. Reynolds, "'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications," *Geomorphology*, vol. 179, pp. 300–314, 2012.

[33] S. Thrun and J. J. Leonard, "Simultaneous localization and mapping," in *Springer handbook of robotics*, Springer Berlin Heidelberg, 2008, pp. 871–889.

[34] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[35] P. Moulon, P. Monasse, and R. Marlet, "Adaptive structure from motion with a contrario model estimation," in *Asian Conference on Computer Vision*, Springer Berlin Heidelberg, 2012, pp. 257–270.

[36] P. Moulon, P. Monasse, R. Marlet, *et al.*, *OpenMVG. An Open Multiple View Geometry library.* `https://github.com/openMVG/openMVG`.

[37] P. Moulon, P. Monasse, and R. Marlet, "Global fusion of relative motions for robust, accurate and scalable Structure from Motion," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3248–3255.

[38] O. Enqvist, F. Kahl, and C. Olsson, "Non-sequential structure from motion," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, IEEE, 2011, pp. 264–271.

[39] K. Cornelis, F. Verbiest, and L. Van Gool, "Drift detection and removal for sequential structure from motion algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1249–1259, 2004.

[40] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, no. 2, pp. 164–168, 1944.

[41] M. Lourakis and A. A. Argyros, "Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment?" In *Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005.*, IEEE, vol. 2, 2005, pp. 1526–1531.

[42] D. Sibley, C. Mei, I. D. Reid, and P. Newman, "Adaptive relative bundle adjustment.," in *Robotics: Science and systems*, vol. 32, 2009, p. 33.

[43] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001.*, IEEE, vol. 1, 2001, pp. I–I.

[44] Itseez. (2018). OpenCV, [Online]. Available: `http://opencv.org/`.

[45] W. Kabsch, "A discussion of the solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 34, no. 5, pp. 827–828, 1978.

[46] A. El-Rabbany, *Introduction to GPS: The Global Positioning System*. Artech house, 2002, p. 22.

[47] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[48] S. W. Thomas, "Decomposing a matrix into simple transformations," *Graphics Gems II*, J. Arvo, Ed., pp. 320–323, 1991. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/B9780080507545500694`.

[49] R. N. Goldman, "Recovering the data from the transformation matrix," *Graphics Gems II*, J. Arvo, Ed., pp. 324–331, 1991. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/B9780080507545500700`.

[50] J.-P. Vert, K. Tsuda, and B. Schölkopf, "A primer on kernel methods," *Kernel Methods in Computational Biology*, pp. 35–70, 2004.

[51] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[52] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.

[53] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe, "Full-resolution residual networks for semantic segmentation in street scenes," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 3309–3318. [Online]. Available: `https://doi.org/10.1109/CVPR.2017.353`.

[54] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distribution," *Bulletin of the Calcutta Mathematical Society*, 1943.

[55] Google. (2018). Android Location API Reference: getAccuracy(), [Online]. Available: `https://developer.android.com/reference/android/location/Location.html#getAccuracy()`.

[56] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, IEEE, vol. 2, 1999, pp. 1150–1157.

[57] OpenMVG authors. (2018). OpenMVG SfM documentation, [Online]. Available: `https://openmvg.readthedocs.io/en/latest/software/SfM/SfM/#openmvg-sfm-pipelines`.

[58] J. D. Gibbons and S. Chakraborti, "Nonparametric statistical inference," in *International Encyclopedia of Statistical Science*, M. Lovric, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 977–979. [Online]. Available: https://doi.org/10.1007/978-3-642-04898-2_420.

[59] C. Croux and C. Dehon, "Influence functions of the Spearman and Kendall correlation measures," *Statistical Methods & Applications*, vol. 19, no. 4, pp. 497–515, Nov. 2010. [Online]. Available: https://doi.org/10.1007/s10260-010-0142-z.

Appendix 1. Additional information on experiments

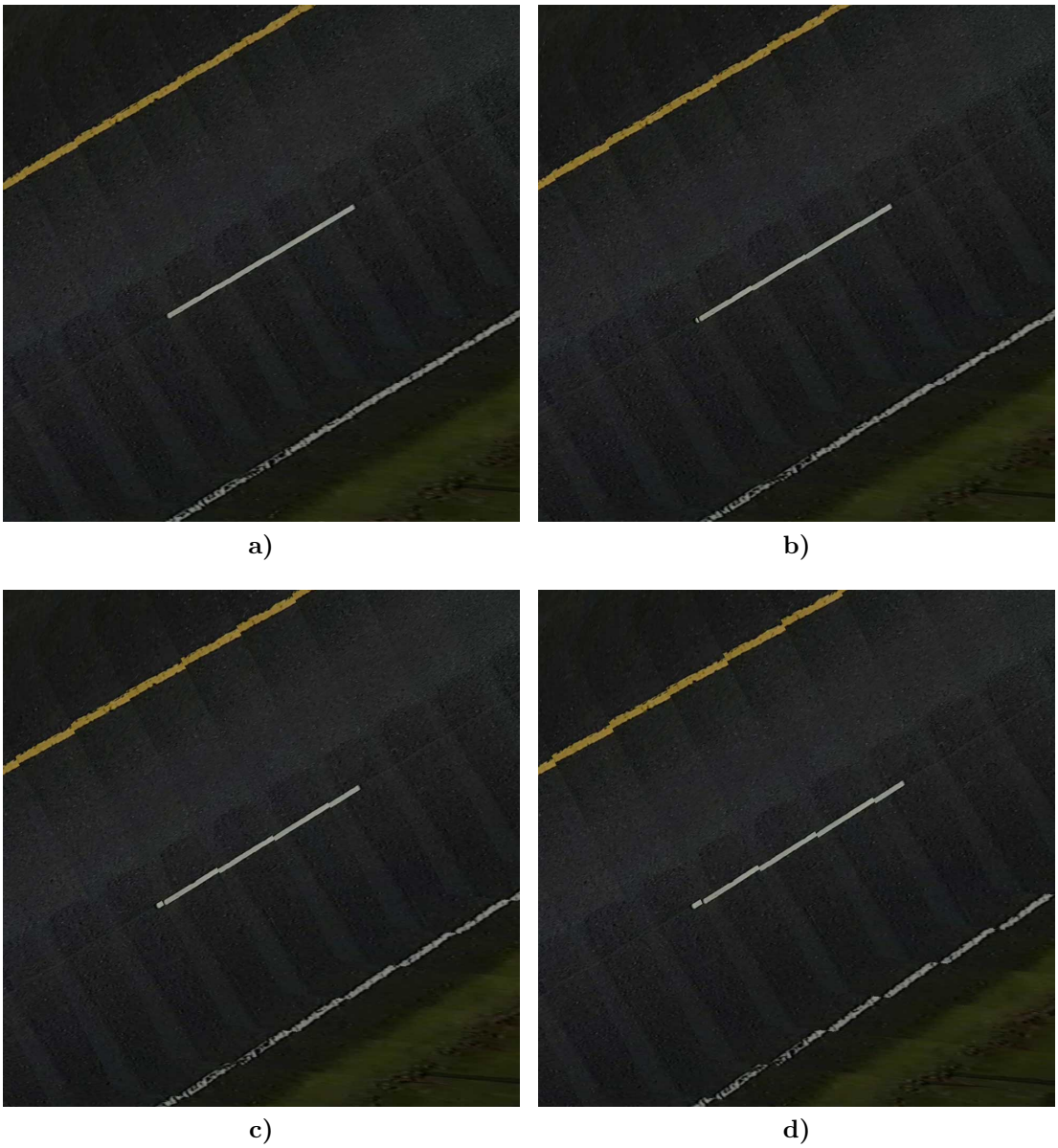
Table A1.1. Parameters used for the SIFT feature descriptor.

Name	Explanation	Value
octaves	number of progressively downsized versions of the original image	4
scales	number of Gaussian blurring stages for each octave	5
edge threshold	threshold used to filter out edge-like features	20.0
peak threshold	minimum contrast threshold	0.001

Table A1.2. Parameters used for the georegistration of 3D reconstructions. The parameters regarding the iterative method and RANSAC refer to the robust estimation of the affine transformation **T** described in Section 4.3.1.

Name	Explanation	Value
σ	standard deviation of the Gaussian kernel for GPS altitude filtering	4.0
w	size of the Gaussian kernel for GPS altitude filtering	21
nMaxIterI	maximum number of iterations used in the iterative method	10
inlierRANSAC	RANSAC inlier distance threshold	5.0 m
nIterRANSAC	number of RANSAC iterations	4000
lineThreshold	inlier distance threshold used to determine the linearity of a segment	5.0 m

Appendix 2. Figures on orthophoto errors

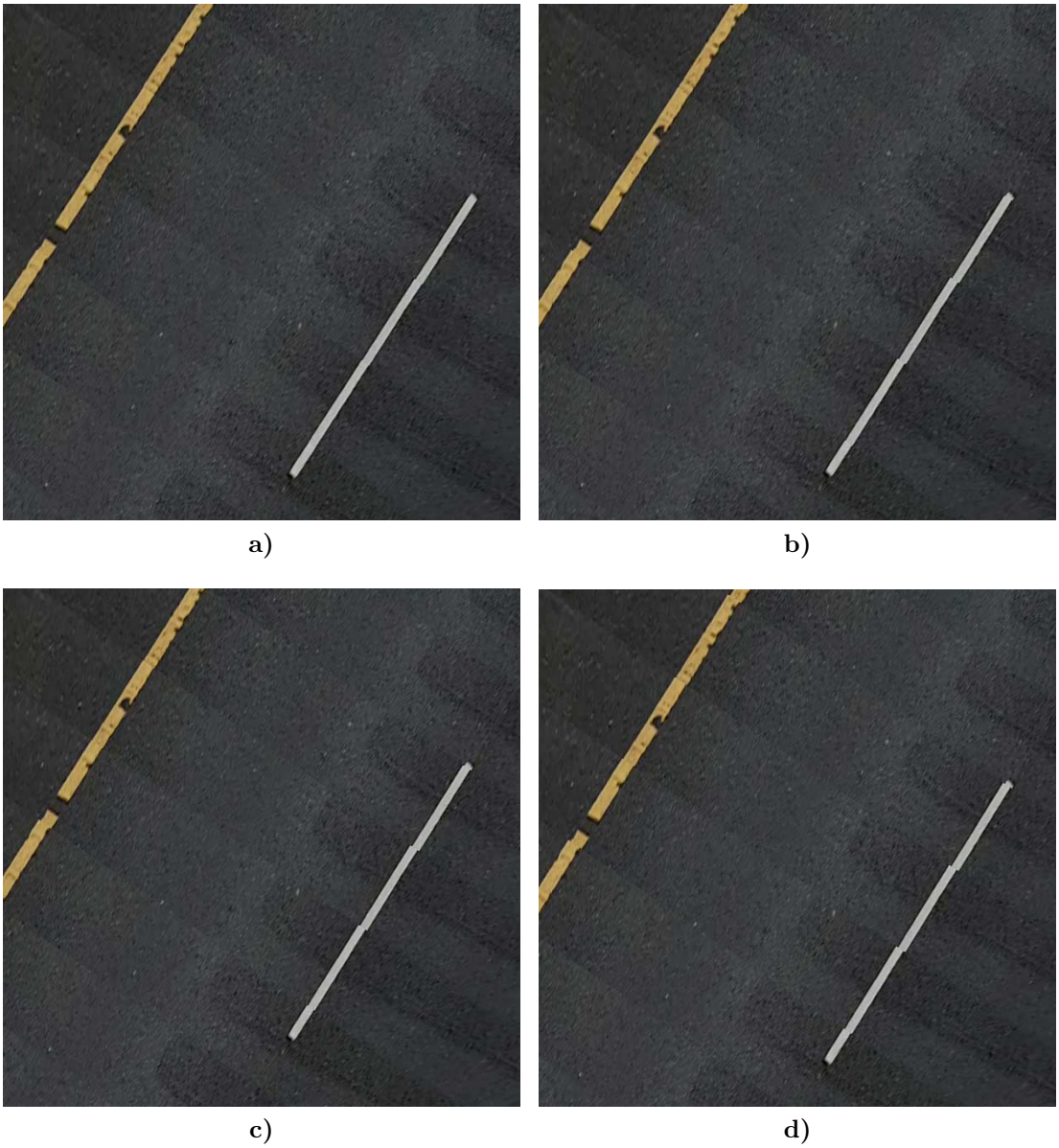


a) **b)**

c) **d)**

Figure A2.1. Impact of camera–road orientation θ estimation error on an orthophoto composite: a) No error; b) Error of 0.5°; c) Error of 1.0°; d) Error of 1.5°.

Appendix 2.

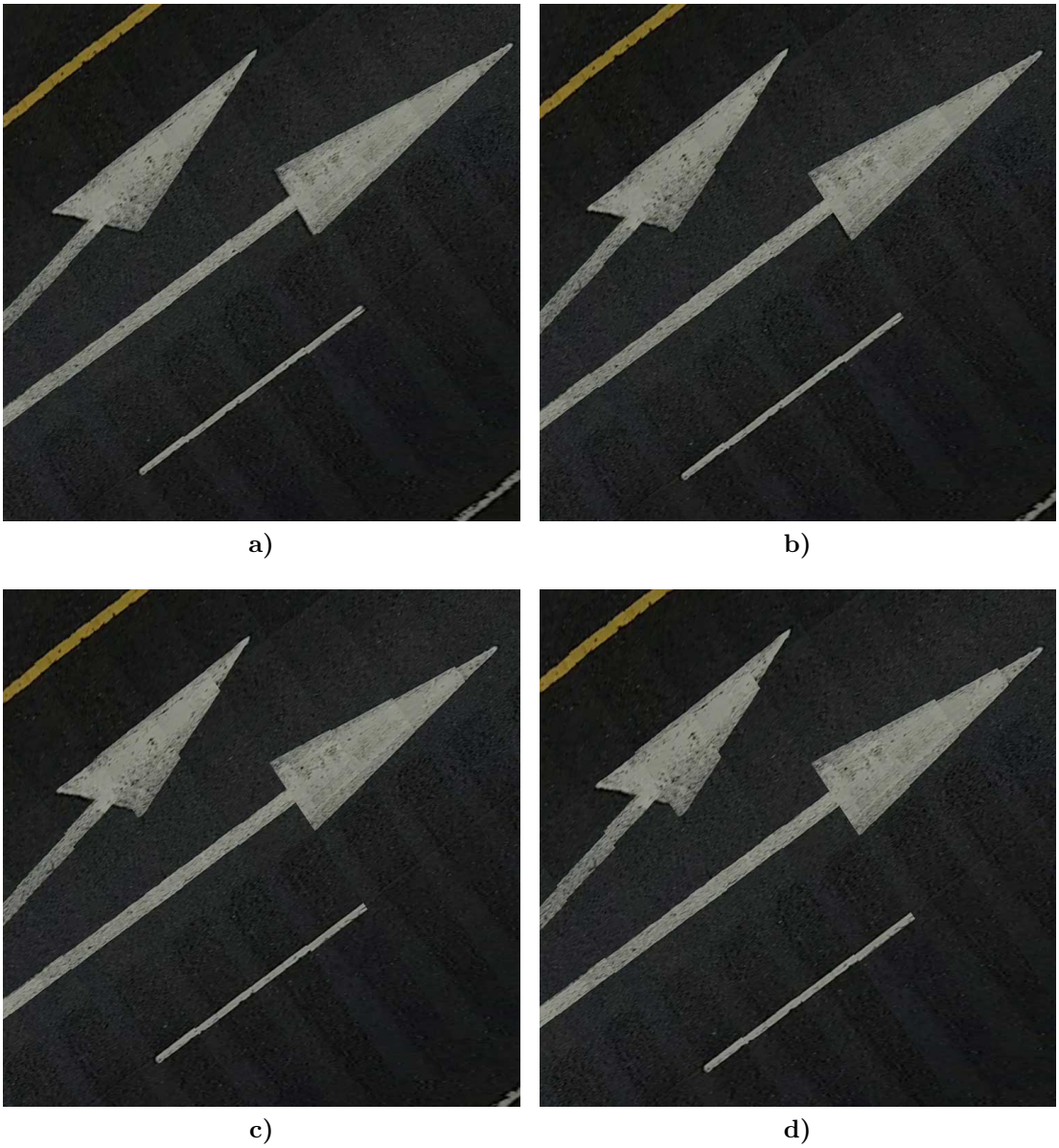


a) **b)**

c) **d)**

Figure A2.2. Impact of camera heading deviation $\Delta\gamma$ estimation error on an orthophoto composite: a) No error; b) Error of 0.5°; c) Error of 0.75°; d) Error of 1.0°.

Appendix 2.



a) **b)**

c) **d)**

Figure A2.3. Impact of camera height d_p estimation error on an orthophoto composite: a) No error; b) Error of 10 cm; c) Error of 20 cm; d) Error of 30 cm.